МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: "*Розробка платформи для віртуальних виставок у мета-всесвіті*"

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи КБ-21-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»



Біленка Івана Івановича

(підпис) (прізвище, ім'я, по батькові)

Науковий керівник 

Редич Олександр Володимирович, к.е.н., доцент

(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми 

Омельчук Антон Анатолійович, к.т.н.

(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних технологій і систем протокол від «13» травня 2025 № 18

Завідувач кафедри 

Філоненко Михайло Миколайович, к.ф.-м.н., доцент

(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2025

Реєстрація № 57 26.05.2025 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 26.05.2025 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 4 (добре) 04.06.2025 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ В.М. Франчук
(підпис)

Члени комісії: _____ М.М. Філоненко
(підпис)

_____ С.М. Грищенко
(підпис)

_____ А.А. Омельчук
(підпис)

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	**від 0 до 30**	26		Редич О.В. Франчук В.М. Філоненко М.М. Грищенко С.М. Омельчук А.А. Суховецька Л.П.
2. Дотримання вимог щодо оформлення	**від 0 до 10**	8		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	**від 0 до 5**	4		
4. Рецензент	**від 0 до 5**	4		
РАЗОМ БАЛІВ	**від 0 до 50**	42		

Оцінювання кваліфікаційної роботи науковим керівником - 41 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100- бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
83	B	4 (добре)	04.06.2025	Франчук В.М. Філоненко М.М. Грищенко С.М. Омельчук А.А. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	F X	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології (за галузями)»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«15» листопада 2024

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Біленку Івану Івановичу

1. Тема кваліфікаційної роботи
«Розробка платформи для віртуальних виставок у мета-всесвіті»
Затверджена наказом ректора від «13»листопада 2024 № 1717
2. Строк здачі закінченої роботи *«5» травня 2025*
3. Мета роботи: *розробка платформи для проведення віртуальних виставок у мета-всесвіті з використанням сучасних веб-технологій та 3D-візуалізації.*

Об'єкт дослідження: *мета-всесвіти та цифрові платформи для онлайн-взаємодії, що становлять технологічну основу для створення віртуальних виставок та експозицій.*

Предмет дослідження: *технології реалізації віртуального простору, зокрема 3D-середовища, інструменти для створення інтерактивного контенту.*

Орієнтовна база дослідження: *існуючі системи віртуального простору, 3D-середовища та мета-всесвіти.*

4. Вхідні дані до роботи: *відкриті дані та документація по інструментарію реалізації віртуальних просторів, 3D-середовищ.*
5. Перелік графічного матеріалу : *рисунки інтерфейсів додатку, структурні схеми, алгоритми, лістинги програм.*
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ ВІРТУАЛЬНИХ СЕРЕДОВИЩ У МЕТА-ВСЕСВІТІ

1.1 Поняття мета-всесвіту, історія та сучасний стан розвитку

1.2 Огляд технологій для реалізації 3D-виставок

1.3 Аналіз існуючих платформ для проведення віртуальних заходів

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПЛАТФОРМИ ДЛЯ ВІРТУАЛЬНИХ ВИСТАВОК

2.1 Архітектура платформи та модулі системи

2.2 Опис середовища для 3D-візуалізації та інтерфейсів

2.3 Технології інтеграції мультимедійного контенту та користувацької взаємодії

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ ПЛАТФОРМИ

3.1 UI/UX дизайн веб-інтерфейсу

3.2 Реалізація функціоналу віртуальної навігації

3.3 Тестування платформи, оцінка продуктивності та економічного ефекту впровадження

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	до 01.11.2024	«01» 11 2024 р.
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	30.01.2025	«30» 01 2025 р.
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	28.02.2025	«28» 02 2025 р.
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	30.04.2025	«30» 04 2025 р.
5	**Висновки**	05.05.2025	«05» 05 2025 р.
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	25.05.2025	«25» 05 2025 р.
7	Отримання відгуку керівника	25.05.2025	«25» 05 2025 р.
8	Отримання зовнішньої рецензії	28.05.2025	«28» 05 2025 р.

8. Дата видачі завдання *«15» листопада 2024 р.*

9. Керівник кваліфікаційної роботи _____ Редич О.В.
(підпис)

10. Керівник освітньо-професійної програми _____ Омельчук А.
(підпис)

11. Завдання до виконання отримав, здобувач вищої освіти _____ Біленко І.І.
(підпис)

АНОТАЦІЯ

Біленко І.І.

“Розробка платформи для віртуальних виставок у мета-всесвіті”

Метою роботи є створення платформи для проведення віртуальних виставок у мета-всесвіті з використанням сучасних веб-технологій та засобів 3D-візуалізації. У роботі проаналізовано поняття мета-всесвіту, розглянуто сучасні платформи для віртуальної взаємодії, а також досліджено інструменти реалізації інтерактивного 3D-контенту. Запропонована архітектура платформи включає модулі для побудови віртуального середовища, мультимедійної інтеграції та зручного веб-інтерфейсу з функціоналом навігації. Реалізовано основні компоненти: UI/UX дизайн, підтримку мультимедійного контенту та механізми інтерактивної взаємодії користувачів. Результати роботи можуть бути використані для організації віртуальних заходів у сфері освіти, бізнесу або культури.
Кваліфікаційна робота містить 48 сторінок, 7 рисунків, список використаних джерел з 40 найменувань, 4 додатків.

Ключові слова: мета-всесвіт, віртуальна виставка, 3D-візуалізація, веб-платформа, інтерактивний контент, UI/UX.

Рік Виконання кваліфікаційної роботи: 2024-2025
Рік захисту роботи: 2025

ABSTRACT

Bilenko I.

"Development of a Platform for Virtual Exhibitions in the Metaverse"

The aim of this work is to develop a platform for hosting virtual exhibitions in the metaverse using modern web technologies and 3D visualization tools. The paper analyzes the concept of the metaverse, reviews current platforms for virtual interaction, and explores tools for implementing interactive 3D content. The proposed platform architecture includes modules for building virtual environments, multimedia integration, and a user-friendly web interface with navigation functionality. The main components were implemented: UI/UX design, multimedia content support, and mechanisms for interactive user engagement. The results of the work can be used to organize virtual events in the fields of education, business, or culture.
The qualification paper contains 48 pages, 7 figures, a list of used sources with 40 items, and 4 appendices.

Keywords: metaverse, virtual exhibition, 3D visualization, web platform, interactive content, UI/UX.

Year of completion: 2024-2025
Year of defense: 2025

ЗМІСТ

ВСТУП **3**

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ ВІРТУАЛЬНИХ СЕРЕДОВИЩ У МЕТА-ВСЕСВІТІ **6**

1.1 Поняття мета-всесвіту, історія та сучасний стан розвитку 6

1.2 Огляд технологій для реалізації 3D-виставок 8

1.3 Аналіз існуючих платформ для проведення віртуальних заходів 12

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПЛАТФОРМИ ДЛЯ ВІРТУАЛЬНИХ ВИСТАВОК **18**

2.1 Архітектура платформи та модулі системи 18

2.2 Опис середовища для 3D-візуалізації та інтерфейсів 24

2.3 Технології інтеграції мультимедійного контенту та користувацької взаємодії 29

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ ПЛАТФОРМИ **36**

3.1 UI/UX дизайн веб-інтерфейсу 36

3.2 Реалізація функціоналу віртуальної навігації 41

3.3 Тестування платформи, оцінка продуктивності та економічного ефекту впровадження 45

ВИСНОВКИ **51**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **53**

ДОДАТКИ **58**

ВСТУП

Віртуальні виставки та цифрові експозиції стають все більш актуальними в умовах тотальної цифровізації суспільства та трансформації традиційних форм культурної взаємодії. Розвиток технологій віртуальної та доповненої реальності, разом із становленням концепції мета-всесвіту, відкриває безпрецедентні можливості для створення нових форматів представлення експозицій, творів мистецтва та інших об'єктів культурної спадщини. Технології мета-всесвіту дозволяють подолати фізичні обмеження традиційних виставкових просторів, забезпечуючи глобальну доступність, інтерактивність та персоналізацію користувацького досвіду.

Актуальність розробки платформи для віртуальних виставок у мета-всесвіті зумовлена кількома факторами. По-перше, пандемія COVID-19 продемонструвала вразливість традиційних форматів проведення виставок та необхідність розвитку цифрових альтернатив. По-друге, зростає глобальний попит на віртуальні культурні та освітні заходи, що забезпечують доступ до контенту незалежно від географічного розташування користувачів. По-третє, формується новий сегмент цифрового мистецтва та NFT, що потребує інноваційних платформ для презентації та взаємодії. Крім того, в Україні та світі спостерігається зростання галузі віртуальних подій, що підтверджується статистикою збільшення інвестицій у відповідні технологічні рішення та платформи.

Об'єктом дослідження є мета-всесвіт та цифрові платформи для онлайн-взаємодії, що становлять технологічну основу для створення віртуальних виставок та експозицій. Мета-всесвіт як концепція постійно діючого віртуального простору з власною економікою, соціальними взаємодіями та культурним контентом представляє собою нову парадигму цифрового досвіду, яка трансформує способи взаємодії користувачів з контентом та один з одним.

Предметом дослідження є технології реалізації віртуального простору, зокрема 3D-середовище, веб-інтерфейси, інструменти для створення

інтерактивного контенту. Це включає як технічні аспекти (рендеринг 3D-графіки, мережева синхронізація, інтеграція різних типів медіа), так і аспекти користувацького досвіду (інтерфейси, взаємодія, доступність).

Метою даної роботи є розробка платформи для проведення віртуальних виставок у мета-всесвіті з використанням сучасних веб-технологій та 3D-візуалізації. Така платформа має забезпечувати повний цикл створення, публікації та відвідування віртуальних виставок, з підтримкою різних типів контенту, навчального середовища, соціальних взаємодій та бізнес-моделей.

Для досягнення поставленої мети визначено наступні завдання:

1. Проаналізувати сучасний стан розвитку технологій мета-всесвіту та існуючих платформ для віртуальних виставок, виявити їх обмеження та можливості для вдосконалення.
2. Розробити архітектуру платформи для віртуальних виставок, визначити її ключові модулі та функціональні компоненти.
3. Спроектувати середовище для 3D-візуалізації, що забезпечує високоякісне відображення експонатів та віртуальних просторів.
4. Розробити технології інтеграції різних типів мультимедійного контенту та механізми користувацької взаємодії.
5. Створити веб-інтерфейс платформи з урахуванням принципів UX/UI дизайну та вимог доступності.
6. Реалізувати функціонал віртуальної навігації для різних режимів перегляду (десктоп, мобільний, VR/AR).
7. Провести тестування розробленої платформи та оцінку її продуктивності на прикладі інтеграції навчальних ресурсів Oracle Academy, Cisco Academy.

Методологія дослідження базується на комплексному підході, що включає аналіз наукової літератури та технічної документації, порівняльний аналіз існуючих рішень, проектування та розробку програмного забезпечення з використанням сучасних методів та інструментів, емпіричне тестування та оцінку результатів.

Теоретичною та методологічною основою дослідження є праці в галузі комп'ютерної графіки, веб-розробки, віртуальної та доповненої реальності, а також дослідження в сфері цифрової культури та мистецтва. Використано методи об'єктно-орієнтованого аналізу та проектування, компонентної архітектури, агентного моделювання, а також принципи дизайну, орієнтованого на користувача.

Практичне значення отриманих результатів визначається можливістю їх використання для створення віртуальних виставок різної тематики та масштабу – від персональних галерей до масштабних міжнародних експозицій. Розроблена платформа може бути застосована в культурній сфері (музеї, галереї, культурні центри), освіті (віртуальні екскурсії, інтерактивні навчальні курси), бізнесі (торгові виставки, презентації продуктів), а також у сфері цифрового мистецтва та NFT.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ ВІРТУАЛЬНИХ СЕРЕДОВИЩ У МЕТА-ВСЕСВІТІ

1.1 Поняття мета-всесвіту, історія та сучасний стан розвитку

Поняття мета-всесвіту (метаверсу) являє собою концепцію постійно діючого віртуального простору, де користувачі можуть взаємодіяти один з одним та з цифровими об'єктами через свої аватари, використовуючи технології віртуальної (VR) та доповненої реальності (AR). Сам термін "метавсесвіт" уперше з'явився в науково-фантастичному романі Ніла Стівенсона "Snow Crash" (1992), де описувався як віртуальний простір, у якому люди могли взаємодіяти через цифрові аватари. Проте концептуальні основи мета-всесвіту почали формуватися ще раніше, зокрема в роботах кіберпанк-письменників, таких як Вільям Гібсон, який запровадив термін "кіберпростір" у своєму романі "Нейромант" (1984).

Розвиток ідеї мета-всесвіту можна умовно розділити на кілька ключових етапів. Початковий етап (1990-ті – 2000-і роки) характеризувався появою перших багатокористувацьких онлайн-світів, таких як "Second Life" (2003), які дозволяли користувачам створювати віртуальні простори, спілкуватися та обмінюватися цифровими товарами. Цей період став першою практичною реалізацією концепції віртуального простору, хоча технологічні можливості того часу значно обмежували потенціал цих платформ.

Наступний етап (2010-ті роки) ознаменувався значним прогресом у технологіях віртуальної та доповненої реальності. Поява пристроїв VR, таких як Oculus Rift (2016), HTC Vive (2016) та PlayStation VR (2016), зробила віртуальну реальність доступнішою для широкого загалу. Паралельно розвивалися технології AR через смартфони та спеціалізовані пристрої, такі як Microsoft HoloLens (2016).

Сучасний етап розвитку мета-всесвіту (з 2020-х років) характеризується інтенсивною інтеграцією різних технологій та концепцій. Значний поштовх до

розвитку метаверсу відбувся в 2021 році, коли компанія Facebook оголосила про перейменування на Meta Platforms та фокусування на розробці метаверсу. Це рішення спричинило хвилю інвестицій та досліджень у цій галузі з боку інших технологічних гігантів, таких як Microsoft, Google, NVIDIA та Apple.

На сьогоднішній день мета-всесвіт не є єдиною централізованою платформою, а скоріше екосистемою взаємопов'язаних віртуальних просторів, які розвиваються паралельно. Серед ключових платформ можна виділити Horizon Worlds від Meta, Decentraland, The Sandbox, та інші. Ці платформи пропонують різні підходи до реалізації концепції мета-всесвіту, від централізованих корпоративних рішень до децентралізованих платформ на основі блокчейну [1].

Технологічною основою сучасного мета-всесвіту є комбінація VR/AR технологій, хмарних обчислень, штучного інтелекту, технологій розподіленого реєстру (блокчейн) та високошвидкісного інтернет-зв'язку. Важливим аспектом є також розвиток технологій 3D-моделювання та рендерингу, які забезпечують високу якість візуального представлення віртуальних об'єктів та просторів.

Економічний вимір мета-всесвіту включає як традиційні бізнес-моделі (передплата, мікротранзакції, реклама), так і нові форми цифрової економіки, засновані на блокчейні. Ключовими елементами останньої є невзаємозамінні токени (NFT), які дозволяють засвідчувати право власності на цифрові активи, та криптовалюти, які використовуються для здійснення транзакцій у віртуальних просторах.

Соціальний аспект мета-всесвіту передбачає нові форми взаємодії та соціалізації, які виходять за рамки традиційних соціальних мереж. У мета-всесвіті користувачі можуть створювати складні соціальні структури, формувати спільноти на основі спільних інтересів, брати участь у колективних подіях та заходах, а також спільно створювати та модифікувати віртуальні простори.

Культурний вплив мета-всесвіту проявляється у формуванні нових форм мистецтва та творчості, розвитку цифрової моди та віртуальних ідентичностей.

Віртуальні концерти, виставки, перформанси стають невід'ємною частиною культурного ландшафту, створюючи нові можливості для творчого самовираження та сприйняття мистецтва.

Перспективи розвитку мета-всесвіту пов'язані з подальшою інтеграцією фізичного та віртуального світів, розвитком технологій нейроінтерфейсів, удосконаленням методів передачі сенсорних відчуттів та створенням більш реалістичних віртуальних аватарів. Очікується, що мета-всесвіт буде все більше проникати в різні сфери життя, включаючи освіту, медицину, роботу, розваги та соціальну взаємодію.

Водночас, розвиток мета-всесвіту супроводжується низкою викликів та проблем. Серед них – питання захисту персональних даних та приватності, проблема цифрової нерівності та доступності, потенційні психологічні наслідки тривалого перебування у віртуальному просторі, етичні аспекти взаємодії з аватарами інших користувачів, а також проблеми регулювання та встановлення правових норм у мета-всесвіті.

Незважаючи на вказані виклики, потенціал мета-всесвіту для трансформації різних аспектів людського життя є значним. За прогнозами аналітиків, глобальний ринок метаверсу може досягти обсягу в понад $1 трильйон до 2030 року, що свідчить про високі очікування щодо його розвитку та впливу на економіку та суспільство в цілому [2].

Таким чином, мета-всесвіт представляє собою комплексну концепцію, яка поєднує технологічні, економічні, соціальні та культурні аспекти, створюючи передумови для формування нового етапу розвитку цифрового суспільства – суспільства, де фізичний та віртуальний простори все більше інтегруються, створюючи нові форми людського досвіду та взаємодії.

1.2 Огляд технологій для реалізації 3D-виставок

Реалізація 3D-виставок у мета-всесвіті спирається на комплекс взаємопов'язаних технологій, які забезпечують створення імерсивного віртуального середовища та ефективну взаємодію користувачів з експонатами

та один з одним. Розгляньмо ключові технології, їх особливості та роль у побудові 3D-виставкових просторів.

3D-моделювання є фундаментальною технологією для створення віртуальних виставок, оскільки забезпечує відтворення фізичних об'єктів, експонатів та архітектурних просторів у цифровому форматі. Сучасний інструментарій 3D-моделювання представлений низкою програмних рішень різного рівня складності: Blender, Autodesk Maya, 3ds Max, Cinema 4D, ZBrush та інші. Ці інструменти дозволяють створювати високодеталізовані моделі з реалістичними текстурами та властивостями матеріалів. Важливою тенденцією є розвиток технологій фотограмметрії, які дозволяють створювати 3D-моделі на основі серії фотографій реального об'єкта, що спрощує процес оцифрування колекцій музеїв, галерей та інших культурних закладів для їх представлення у віртуальному просторі.

Технології 3D-рендерингу відповідають за перетворення 3D-моделей у реалістичне зображення, яке користувач бачить на екрані. Сучасні рушії рендерингу, такі як Unreal Engine, Unity, V-Ray, Arnold та інші, використовують різні методи обчислення освітлення та відображення поверхонь, включаючи трасування променів (ray tracing), глобальне освітлення (global illumination) та процедурний рендеринг. Особливо важливим аспектом для 3D-виставок є реалістичне відтворення освітлення та тіней, оскільки це значно впливає на сприйняття експонатів. Рендеринг у реальному часі, характерний для ігрових рушіїв, дозволяє створювати інтерактивні виставки, де освітлення та візуальні ефекти можуть динамічно змінюватися відповідно до дій користувача.

Веб-технології для 3D-візуалізації набувають особливого значення, оскільки дозволяють забезпечити доступ до віртуальних виставок через браузер, без необхідності встановлення спеціалізованого програмного забезпечення. Ключовою технологією в цьому контексті є WebGL – JavaScript API для рендерингу інтерактивної 3D-графіки у веб-браузері без використання плагінів. На основі WebGL розроблено низку бібліотек та фреймворків, які спрощують створення 3D-контенту для веб-середовища: Three.js, Babylon.js,

PlayCanvas та інші. Також варто відзначити формат glTF (GL Transmission Format), який став стандартом для передачі 3D-моделей у веб-середовищі, забезпечуючи ефективне стиснення даних та швидке завантаження [3].

Технології віртуальної реальності (VR) дозволяють користувачам повністю зануритися у віртуальний виставковий простір, ізолювавшись від фізичного оточення. Основними компонентами VR є шоломи/окуляри віртуальної реальності (HMD – Head-Mounted Display), такі як Oculus Quest, HTC Vive, Valve Index, PlayStation VR, а також контролери для взаємодії з віртуальними об'єктами. Важливими аспектами VR-технологій для виставок є технології відстеження положення користувача у просторі (6DOF – six degrees of freedom), які забезпечують вільне пересування у віртуальному просторі, та технології відстеження рук та жестів для природної взаємодії з експонатами. Стандарти та фреймворки, такі як OpenXR, WebXR, SteamVR, забезпечують уніфікований інтерфейс для розробки VR-додатків та їх сумісність з різними пристроями.

Технології доповненої реальності (AR) дозволяють накладати віртуальні елементи на реальний світ, створюючи гібридний досвід для відвідувачів виставок. На відміну від VR, AR не ізолює користувача від фізичного оточення, а доповнює його цифровим контентом. Основними технологіями AR для виставок є: маркерна AR, яка використовує спеціальні візуальні маркери для позиціонування віртуальних об'єктів; безмаркерна AR, яка розпізнає особливості фізичного середовища та розміщує віртуальні об'єкти відповідно до них; SLAM (Simultaneous Localization and Mapping) – технологія, яка дозволяє пристрою створювати карту оточуючого простору та визначати своє положення в ньому. Ключовими платформами для розробки AR-додатків є ARKit (Apple), ARCore (Google), Vuforia, AR.js (для веб-AR).

Технології змішаної реальності (MR) представляють наступний етап розвитку, поєднуючи елементи VR та AR та забезпечуючи більш глибоку інтеграцію віртуальних об'єктів з фізичним середовищем. MR-пристрої, такі як Microsoft HoloLens та Magic Leap, використовують складні системи

просторового картографування та розуміння фізичного простору, що дозволяє віртуальним експонатам взаємодіяти з реальними об'єктами, реагувати на освітлення, відкидати тіні та навіть приховуватися за реальними перешкодами. Ця технологія особливо перспективна для створення гібридних виставок, де фізичні та віртуальні експонати співіснують в одному просторі.

Технології взаємодії користувача з віртуальним середовищем включають як традиційні методи (маніпулятори, жести, голосові команди), так і новітні рішення. Технології відстеження руху тіла (motion capture) дозволяють переносити рухи користувача на його аватар у віртуальному просторі. Технології тактильного зворотного зв'язку (haptic feedback) забезпечують відчуття дотику до віртуальних об'єктів через спеціальні рукавички або контролери з вібромоторами. Технології розпізнавання жестів та міміки дозволяють аватарам користувачів відтворювати їхні емоції та жести, що важливо для соціальної взаємодії на віртуальних виставках. Голосове керування та чат-боти на основі штучного інтелекту можуть виконувати роль віртуальних гідів, надаючи інформацію про експонати та відповідаючи на запитання відвідувачів.

Мережеві технології та серверні рішення забезпечують одночасний доступ багатьох користувачів до віртуальної виставки, синхронізацію їхніх дій та інтеракцій. Ключовими технологіями в цьому контексті є WebSockets для забезпечення двостороннього зв'язку між клієнтом та сервером у реальному часі, WebRTC для пірингової комунікації між користувачами (аудіо, відео, дані), та хмарні обчислення для забезпечення масштабованості та балансування навантаження. Для великих віртуальних виставок з багатьма одночасними користувачами використовуються технології шардингу (розподілу віртуального простору на сегменти, які обробляються різними серверами) та інстансингу (створення копій виставкового простору для різних груп відвідувачів).

Технології штучного інтелекту та машинного навчання знаходять різноманітне застосування у віртуальних виставках. Вони використовуються для створення інтелектуальних віртуальних гідів, персоналізації

користувацького досвіду на основі аналізу поведінки та вподобань, генерації динамічного контенту та адаптації виставкового простору. Технології комп'ютерного зору дозволяють аналізувати реальні об'єкти та автоматизувати процес створення їхніх 3D-моделей. Генеративні нейронні мережі можуть створювати нові експонати або модифікувати існуючі, надаючи виставкам інтерактивного та еволюційного характеру.

Технології блокчейну та NFT (невзаємозамінних токенів) забезпечують нові механізми для підтвердження автентичності та права власності на цифрові експонати, а також можливості для монетизації цифрового мистецтва. NFT дозволяють художникам створювати унікальні цифрові твори, які можуть бути куплені, продані або передані, зберігаючи при цьому запис про історію власності та провенанс. Це створює нову економічну модель для віртуальних виставок та галерей, де цифрові експонати можуть мати реальну ринкову вартість.

Технології захисту даних та забезпечення приватності користувачів набувають особливого значення у контексті віртуальних виставок, де може збиратися інформація про поведінку відвідувачів, їхні вподобання та взаємодії. Технології шифрування, анонімізації даних, управління цифровими ідентичностями та контролю доступу до різних частин віртуального простору є важливими компонентами інфраструктури безпеки віртуальних виставок [4].

Таким чином, реалізація 3D-виставок у мета-всесвіті потребує інтеграції широкого спектру технологій, які разом створюють потужну платформу для презентації контенту та взаємодії користувачів. Постійний розвиток цих технологій відкриває нові можливості для створення все більш реалістичних, інтерактивних та імерсивних виставкових просторів, які перевершують можливості фізичних виставок та створюють унікальний досвід для відвідувачів.

1.3 Аналіз існуючих платформ для проведення віртуальних заходів

Ринок платформ для проведення віртуальних заходів активно розвивається, пропонуючи різноманітні рішення для організації віртуальних виставок, конференцій, презентацій та інших подій у цифровому просторі. Аналіз існуючих платформ дозволяє виявити їхні основні характеристики, переваги, обмеження та тенденції розвитку, що є важливим для розробки нової платформи для віртуальних виставок у мета-всесвіті.

Платформа Spatial (spatial.io) є одним з провідних рішень для організації віртуальних виставок та галерей мистецтва, що підтримує як VR, так і веб-доступ. Ключовими особливостями Spatial є мультиплатформність (VR-пристрої, ПК, мобільні пристрої), високоякісне відтворення цифрових творів мистецтва з підтримкою різних форматів (2D, 3D, відео, аудіо), можливість проведення віртуальних екскурсій з гідом у реальному часі та інтеграція з блокчейн-платформами для підтримки NFT. Spatial надає користувачам набір шаблонів виставкових просторів, які можна налаштовувати відповідно до власних потреб, а також можливість імпорту власних 3D-моделей для створення унікальних просторів. Обмеженнями платформи є відносно висока вартість підписки для організаторів, а також обмеження на кількість одночасних користувачів у одному просторі (до 50 для преміум-планів).

Mozilla Hubs представляє собою відкриту веб-платформу для створення віртуальних просторів, доступну через браузер без необхідності встановлення спеціального програмного забезпечення. Ключовими перевагами Mozilla Hubs є низький поріг входу для користувачів (достатньо посилання для доступу до віртуального простору), кросплатформеність (підтримка VR, ПК, мобільних пристроїв), відкритий код та можливість самостійного хостингу. Платформа надає інструменти для завантаження та розміщення 3D-моделей, зображень, відео та аудіо, а також підтримує інтеграцію з популярними 3D-форматами (glTF, OBJ, FBX) [5]. Обмеженнями Mozilla Hubs є відносно простий візуальний стиль, обмежені можливості для налаштування інтерфейсу користувача та обмеження на кількість одночасних користувачів у одному просторі (до 25-30 для стандартної конфігурації).

VirBELA спеціалізується на створенні віртуальних просторів для проведення бізнес-заходів, включаючи виставки, конференції та мережеві події. Платформа пропонує готові віртуальні кампуси та виставкові центри з різними функціональними зонами (виставкові зали, конференц-зали, аудиторії, зони для нетворкінгу). Ключовими перевагами VirBELA є висока масштабованість (підтримка кількох тисяч одночасних користувачів), розвинені інструменти для взаємодії (голосовий чат з просторовим звуком, текстовий чат, спільний перегляд презентацій), можливість налаштування брендингу та інтеграція з бізнес-інструментами. Обмеженнями є відносно спрощена графіка, обмежені можливості для індивідуального налаштування архітектури просторів та орієнтація переважно на корпоративний сегмент з відповідним ціноутворенням.

Engage VR позиціонується як платформа для освітніх та корпоративних заходів у віртуальній реальності, але також пропонує можливості для організації виставок та презентацій. Engage надає доступ до бібліотеки готових 3D-просторів та моделей, інструменти для створення власного контенту та можливість запису сесій для подальшого перегляду. Ключовими перевагами платформи є висока якість візуалізації (на основі рушія Unreal Engine), підтримка широкого спектру VR-пристроїв, інструменти для створення інтерактивних симуляцій та сценаріїв [6].

AltspaceVR, придбаний Microsoft у 2017 році, є соціальною VR-платформою з можливостями для проведення різноманітних заходів, включаючи виставки, презентації та концерти. Платформа пропонує низку готових шаблонів просторів, інструменти для створення власного контенту (World Editor) та SDK для розробників. Ключовими перевагами AltspaceVR є підтримка широкого спектру VR-пристроїв (включаючи Windows Mixed Reality, Oculus, HTC Vive), добре розвинені соціальні функції (аватари з мімікою, жести, емоції), інтеграція з Microsoft Teams та відносно низький поріг входу для користувачів. Обмеженнями є обмежена кількість одночасних користувачів у одному просторі (до 50-80 залежно від типу події), стилізована графіка та певні обмеження на імпорт власних 3D-моделей.

Frame VR пропонує легкий спосіб створення віртуальних просторів для виставок, що доступні через веб-браузер. Платформа відрізняється мінімалістичним інтерфейсом та фокусом на простоті використання. Користувачі можуть швидко створювати 3D-простори, імпортувати моделі, зображення, відео та інший контент через інтуїтивний редактор. Ключовими перевагами Frame VR є низький поріг входу (не потрібне спеціальне обладнання або програмне забезпечення), швидке розгортання просторів та інтеграція з популярними веб-інструментами. Обмеженнями є обмежені можливості для налаштування інтерфейсу, відносно проста графіка та обмеження на кількість одночасних користувачів у безкоштовній версії.

The Sandbox є платформою на основі блокчейну, що дозволяє користувачам створювати, володіти та монетизувати віртуальні активи та досвід. Хоча The Sandbox має більш ігрову орієнтацію, вона все частіше використовується для проведення віртуальних виставок та подій, особливо у сфері цифрового мистецтва та NFT. Ключовими особливостями The Sandbox є можливість придбання віртуальної землі (LAND), створення та продажу віртуальних активів (ASSET) з використанням вбудованого редактора VoxEdit, розробка інтерактивних сценаріїв через Game Maker та власна криптовалюта SAND для проведення транзакцій. Обмеженнями є висока вартість входу для організаторів (ціни на віртуальну землю), вокселний стиль графіки, що може не підходити для всіх типів виставок, та технічні обмеження на складність сцен.

NVIDIA Omniverse представляє платформу для спільної роботи над 3D-проектами в реальному часі, яка також може бути використана для створення високоякісних віртуальних виставок. Omniverse відрізняється від інших платформ фокусом на фотореалістичному рендерингу та сумісності з професійними інструментами 3D-моделювання. Ключовими перевагами Omniverse є підтримка формату Universal Scene Description (USD) для обміну 3D-даними між різними програмами, використання технології RTX для рендерингу з трасуванням променів у реальному часі, потужні інструменти для симуляції фізики та можливість інтеграції з AI-сервісами NVIDIA. Обмеженнями є високі

вимоги до апаратного забезпечення, орієнтація на професійних користувачів та висока вартість ліцензій для комерційного використання.

Roblox, хоча найбільше відомий як платформа для створення та гри в ігри, все частіше використовується для організації віртуальних виставок, концертів та інших подій. Платформа надає потужні інструменти для створення віртуальних просторів (Roblox Studio) з можливістю програмування інтерактивних елементів за допомогою мови Lua. Ключовими перевагами Roblox є величезна активна аудиторія (особливо серед молодого покоління), мультиплатформеність, добре розвинена економіка з можливістю монетизації контенту та високий рівень масштабованості (підтримка сотень одночасних користувачів у одному просторі). Обмеженнями є специфічний візуальний стиль, що може не підходити для певних типів професійних виставок, певні обмеження на імпорт зовнішніх моделей та орієнтація на більш ігровий досвід.

Facebook Horizon (Meta Horizon Worlds) представляє собою соціальну VR-платформу від Meta, що дозволяє користувачам створювати віртуальні простори та взаємодіяти в них. Ключовими перевагами Horizon є інтеграція з екосистемою Meta (Facebook, Instagram), орієнтація на соціальну взаємодію з реалістичними аватарами та жестами, інтуїтивні інструменти для створення контенту безпосередньо у VR та потенційно величезна користувацька база. Обмеженнями є обмежена доступність (переважно для пристроїв Oculus), обмеження на складність створюваних просторів та відносна новизна платформи з не всіма реалізованими функціями.

Аналіз існуючих платформ для проведення віртуальних заходів дозволяє виділити кілька ключових тенденцій та викликів у цій галузі. Спостерігається рух до підвищення реалістичності та імерсивності віртуальних просторів, розширення можливостей для соціальної взаємодії, інтеграція економічних механізмів на основі блокчейну та NFT, а також зниження порогу входу для користувачів через підтримку веб-доступу та мультиплатформеності [7]. Водночас, виклики включають проблеми масштабованості для забезпечення одночасної присутності великої кількості користувачів, балансування між якістю

візуалізації та доступністю на різних пристроях, а також створення інтуїтивних інструментів для розробки контенту неспеціалістами.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПЛАТФОРМИ ДЛЯ ВІРТУАЛЬНИХ ВИСТАВОК

2.1 Архітектура платформи та модулі системи

Архітектура платформи для віртуальних виставок у мета-всесвіті є ключовим аспектом, що визначає її функціональність, продуктивність, масштабованість та користувацький досвід. Ефективна архітектурна концепція повинна враховувати специфічні вимоги віртуальних виставок, забезпечувати гнучкість для різних типів контенту та сценаріїв використання, а також створювати основу для подальшого розвитку та розширення функціональності системи. На Рис. 2.1 представлена діаграма варіантів використання платформи, що відображає основні ролі користувачів та їх взаємодію з системою.



Рис. 2.1 - Діаграма варіантів використання (Use Case Diagram)

Запропонована архітектура платформи базується на мікросервісному підході, який забезпечує гнучкість, масштабованість та можливість незалежного розвитку окремих компонентів системи. На відміну від монолітної архітектури, де всі функції реалізовані в рамках єдиного додатку, мікросервісна архітектура розбиває систему на набір невеликих, автономних сервісів, кожен з яких відповідає за конкретну функціональну область. Такий підхід дозволяє оптимізувати розробку та розгортання, покращити відмовостійкість системи та забезпечити більш ефективне використання ресурсів. На Рис. 2.2 представлена діаграма компонентів платформи, що відображає взаємозв'язки між основними модулями системи.

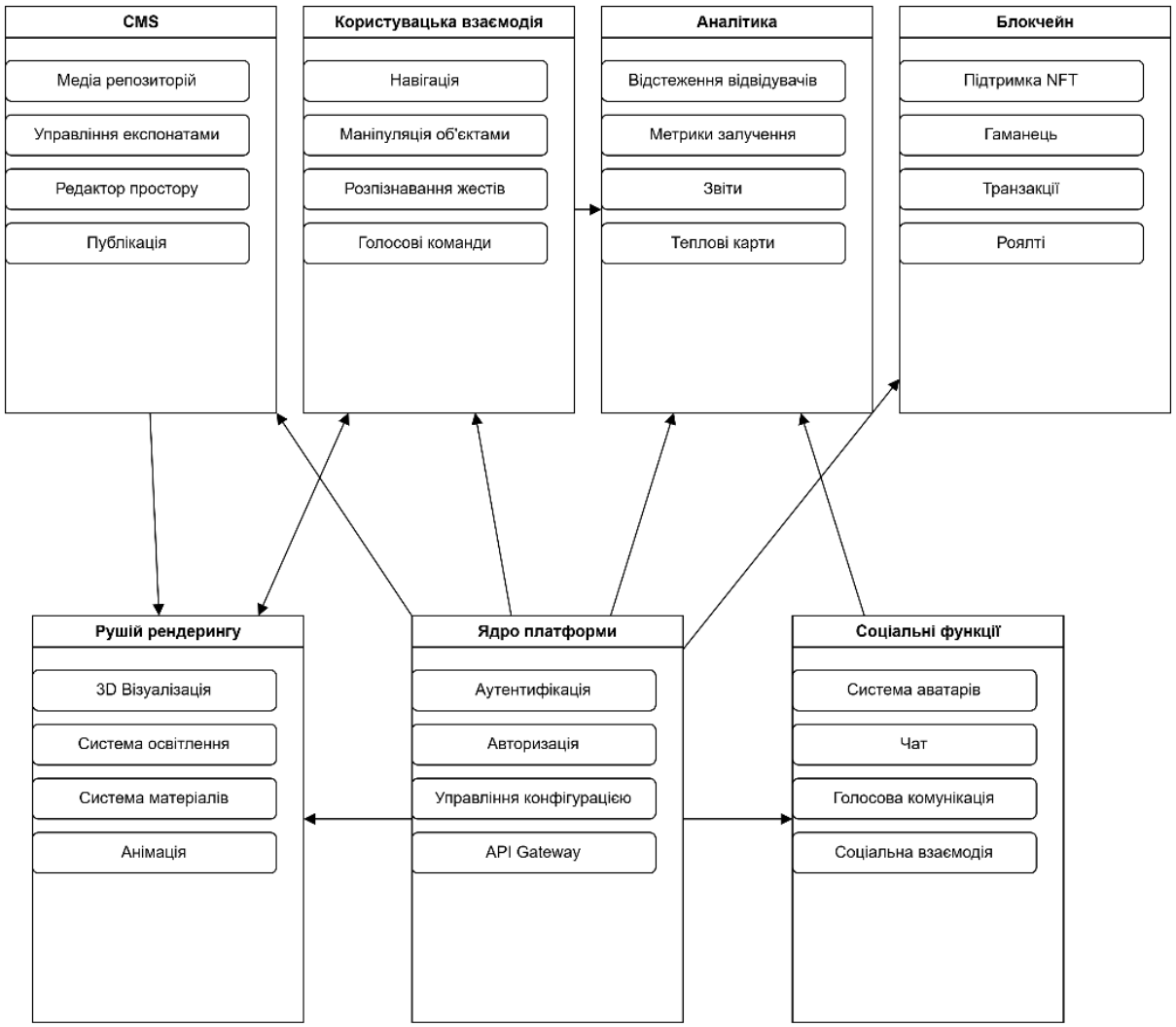


Рис. 2.2 - Діаграма компонентів (Component Diagram)

Архітектура платформи включає три основні рівні: клієнтський рівень, рівень сервісів та рівень даних. Клієнтський рівень представлений веб-інтерфейсом, мобільними додатками та VR/AR-клієнтами, які забезпечують доступ користувачів до платформи з різних пристроїв. Рівень сервісів включає набір мікросервісів, кожен з яких відповідає за конкретну функціональність системи. Рівень даних забезпечує зберігання та управління даними платформи, включаючи реляційні та NoSQL бази даних, системи зберігання файлів та блокчейн-сховища для NFT та інших криптоактивів.

Центральним елементом архітектури є ядро платформи (Core Platform), яке забезпечує координацію роботи різних мікросервісів, управління конфігурацією системи, аутентифікацію та авторизацію користувачів, а також моніторинг стану системи. Ядро платформи реалізує основні бізнес-процеси, такі як створення та публікація виставок, управління користувачами та контентом, обробка платежів та інтеграція з зовнішніми системами. Важливими компонентами ядра є API Gateway, який забезпечує єдину точку входу для клієнтських додатків, та Service Registry, який відповідає за реєстрацію та виявлення мікросервісів у системі.

Один з ключових модулів системи — 3D Rendering Engine — відповідає за візуалізацію тривимірних об'єктів та просторів у реальному часі. Цей модуль забезпечує рендеринг 3D-моделей з високою якістю, управління освітленням та тінями, застосування текстур та матеріалів, а також реалізацію спеціальних візуальних ефектів [8]. Для досягнення оптимальної продуктивності на різних пристроях, модуль підтримує адаптивний рендеринг, який автоматично регулює рівень деталізації моделей та якість візуалізації залежно від потужності пристрою та пропускної здатності мережі. Важливою особливістю модуля є підтримка як клієнтського рендерингу (для високопродуктивних пристроїв), так і серверного рендерингу (для пристроїв з обмеженими ресурсами).

Модуль Spatial Interaction забезпечує взаємодію користувачів з віртуальним простором та об'єктами в ньому. Він реалізує функції навігації у просторі (переміщення, телепортація, зміна масштабу), маніпуляції об'єктами (вибір, переміщення, обертання, масштабування), а також взаємодії з інтерактивними елементами виставки (запуск анімацій, перегляд медіаконтенту, активація інформаційних панелей). Модуль підтримує різні методи введення, включаючи традиційні (клавіатура, миша, тачскрін) та специфічні для VR/AR (контролери руху, відстеження рук та жестів, голосові команди).

Модуль Multi-user Experience відповідає за реалізацію соціальних функцій платформи, що дозволяють користувачам спільно відвідувати віртуальні виставки та взаємодіяти один з одним. Ключовими компонентами цього модуля є системи управління аватарами, синхронізації стану віртуального простору між користувачами, комунікації (голосовий та текстовий чат, система жестів та емоцій) та соціальної взаємодії (формування груп, спільні активності, система репутації). Модуль реалізує різні моделі присутності користувачів, від повністю синхронної (всі користувачі бачать одне й те саме в реальному часі) до асинхронної (можливість залишати повідомлення та взаємодіяти з записами попередніх відвідувачів).

Модуль Content Management System (CMS) забезпечує створення, редагування, публікацію та управління контентом виставок. Він надає кураторам та організаторам виставок інструменти для завантаження та організації 3D-моделей, зображень, відео, аудіо та текстових матеріалів, створення та налаштування віртуальних просторів, розміщення експонатів та інформаційних матеріалів, а також налаштування інтерактивних елементів та сценаріїв взаємодії. CMS підтримує різні рівні доступу та робочі процеси для колаборативної роботи над виставками, версіонування контенту, а також попередній перегляд та тестування виставок перед публікацією.

Модуль Analytics and Insights забезпечує збір, аналіз та візуалізацію даних про взаємодію користувачів з виставками. Він відстежує такі метрики, як кількість відвідувачів, час перебування, маршрути переміщення, взаємодії з

експонатами, конверсію (наприклад, перехід на сайт художника, придбання твору) та інші показники ефективності виставки. Модуль надає організаторам виставок інструменти для аналізу цих даних, виявлення патернів поведінки користувачів, оцінки ефективності різних елементів виставки та прийняття обґрунтованих рішень для оптимізації користувацького досвіду [9]. Важливим аспектом модуля є дотримання приватності користувачів та відповідності регуляторним вимогам.

Модуль Security and Identity Management забезпечує захист системи та даних користувачів, а також управління цифровими ідентичностями. Він реалізує механізми аутентифікації (багатофакторна аутентифікація, OAuth, SAML), авторизації (рольова модель доступу, атрибутивна модель доступу), шифрування даних, захисту від різноманітних атак та відповідності вимогам приватності (GDPR, CCPA та інші).

Модуль Blockchain and NFT Integration забезпечує взаємодію з блокчейн-платформами та підтримку невзаємозамінних токенів (NFT). Він дозволяє художникам та кураторам створювати, випускати та продавати цифрові твори мистецтва як NFT, забезпечує верифікацію автентичності та провенансу цифрових експонатів, а також реалізує механізми для роялті та інших форм монетизації. Модуль підтримує різні блокчейн-платформи (Ethereum, Solana, Flow та інші) та стандарти NFT (ERC-721, ERC-1155 та інші), а також забезпечує інтеграцію з популярними маркетплейсами NFT.

Модуль Payment and Monetization надає різноманітні механізми для монетизації віртуальних виставок та пов'язаних з ними послуг. Він підтримує різні моделі монетизації, включаючи продаж квитків та підписок, мікротранзакції, спонсорство та рекламу, комісійні з продажу творів мистецтва та інші. Модуль забезпечує обробку платежів через різні платіжні системи (кредитні картки, PayPal, криптовалюти).

Модуль Integration Services забезпечує взаємодію платформи з зовнішніми системами та сервісами. Він реалізує інтеграцію з соціальними мережами (для поширення інформації про виставки та авторизації користувачів), системами

управління відносинами з клієнтами (CRM), маркетинговими платформами, системами управління контентом музеїв та галерей, а також з іншими платформами мета-всесвіту. Важливим аспектом цього модуля є підтримка стандартів інтероперабельності, таких як OpenXR, WebXR, glTF, Universal Scene Description (USD), які забезпечують сумісність з іншими системами та можливість обміну контентом.

Модуль Infrastructure and Operations забезпечує технічну інфраструктуру для розгортання та функціонування платформи. Він включає компоненти для управління хмарними ресурсами (серверами, сховищами, мережами), контейнеризації та оркестрації мікросервісів, автоматизації процесів розгортання та оновлення (CI/CD), моніторингу та управління продуктивністю, резервного копіювання та аварійного відновлення. Модуль забезпечує високу доступність та масштабованість системи, автоматичне масштабування ресурсів відповідно до навантаження, а також географічний розподіл для забезпечення низької затримки для користувачів з різних регіонів.

Модуль Accessibility and Localization забезпечує доступність платформи для користувачів з різними можливостями та потребами, а також її адаптацію до різних мов та культурних контекстів. Він реалізує функції для користувачів з обмеженими можливостями (альтернативні методи навігації та взаємодії, аудіодескрипції, налаштування відображення), підтримку різних мов для інтерфейсу та контенту, адаптацію контенту до різних культурних контекстів, а також забезпечує відповідність стандартам доступності [10].

Модуль Education and Learning забезпечує функціональність для освітніх аспектів віртуальних виставок. Він реалізує такі функції, як створення та проведення віртуальних екскурсій та майстер-класів, інтеграція навчальних матеріалів та квізів, відстеження прогресу навчання, створення освітніх сценаріїв та взаємодій, а також підтримка колаборативного навчання в групах. Модуль може інтегруватися з системами управління навчанням

2.2 Опис середовища для 3D-візуалізації та інтерфейсів

Середовище для 3D-візуалізації є ключовим компонентом платформи для віртуальних виставок у мета-всесвіті, оскільки воно визначає візуальну якість, інтерактивність та загальний користувацький досвід. Запропоноване середовище базується на сучасних технологіях комп'ютерної графіки та взаємодії людини з комп'ютером, забезпечуючи високоякісну візуалізацію 3D-об'єктів, реалістичні ефекти освітлення та інтуїтивні інтерфейси для навігації та взаємодії з експонатами ,частина коду показана на Рис. 2.3 (Додаток В).

```
// Завантаження обраних об'єктів з localStorage
let selectedObjects = [];
try {
  selectedObjects = JSON.parse(localStorage.getItem('selectedObjects')) || [];
} catch (e) {}
console.log('Loaded objects from localStorage:', selectedObjects);

// Масив для збереження mesh-об'єктів для опису
const loadedObjects = [];

// Додаємо всі вибрані об'єкти у кімнату
if (selectedObjects.length > 0) {
  selectedObjects.forEach((obj, i) => {
    const angle = (i / selectedObjects.length) * Math.PI * 2;
    const radius = 20;
    const x = Math.cos(angle) * radius;
    const z = Math.sin(angle) * radius;
    const y = -roomHeight / 2 + 3;

    // Пілар (циліндр)
    const pillarHeight = 6;
    const pillarGeometry = new THREE.CylinderGeometry(2, 2, pillarHeight, 32);
    const pillarMaterial = new THREE.MeshStandardMaterial({ color: 0xf0f0f0 });
    const pillar = new THREE.Mesh(pillarGeometry, pillarMaterial);
    pillar.position.set(x, y + pillarHeight / 2 + 6, z);
    scene.add(pillar);

    // Додаємо об'єкт на пілар трохи вище
    if (obj.url && obj.url.match(/\.(gltf|glb)$/i)) {
      const loader = new THREE.GLTFLoader();
      loader.load(obj.url, (gltf) => {
```

Рис. 2.3 Приклад коду завантаження обєктів

В основі середовища для 3D-візуалізації лежить рушій реального часу (real-time engine), який забезпечує рендеринг тривимірної графіки з високою частотою кадрів, що є критичним для створення плавного та імерсивного досвіду, особливо для користувачів VR-пристроїв. Для реалізації платформи обрано гібридний підхід, що поєднує веб-технології (WebGL, WebXR) для забезпечення широкої доступності через браузер та нативний рендеринг для VR/AR-пристроїв, який забезпечує максимальну продуктивність та якість.

Ядром рушія є бібліотека Three.js, яка надає високорівневий API для роботи з 3D-графікою у веб-середовищі, з додатковими розширеннями для підтримки специфічних функцій платформи.

Рендеринг об'єктів у середовищі базується на сучасній рендеринг-архітектурі, що включає фізично-коректне освітлення, технології глобального освітлення для реалістичного відтворення непрямого освітлення та відбиттів, системи частинок для створення атмосферних ефектів, а також пост-обробку для підвищення візуальної якості. [11].

Матеріали та текстури відіграють ключову роль у реалістичному відтворенні експонатів. Система матеріалів базується на принципах фізично-коректного рендерингу (PBR), де властивості поверхні визначаються набором параметрів, що відповідають фізичним характеристикам реальних матеріалів: базовий колір, металічність, шорсткість, нормальна карта для деталізації поверхні, карта оклюзії навколишнього середовища та карта емісії для самосвітних матеріалів. Така система дозволяє досягти високого рівня реалізму при відтворенні різноманітних матеріалів – від матової глини та дерева до полірованого металу, скла та тканин. Для оптимізації продуктивності впроваджено систему автоматичного зниження роздільної здатності текстур та спрощення матеріалів для об'єктів, що знаходяться на великій відстані від користувача.

Система управління 3D-моделями забезпечує ефективне завантаження, обробку та відображення тривимірних об'єктів з різних джерел. Платформа підтримує широкий спектр форматів 3D-моделей (glTF/GLB, OBJ, FBX, USDZ та інші), з автоматичною конвертацією у оптимізований формат для використання на платформі. Реалізовано потокове завантаження моделей з прогресивною деталізацією (LOD – Level of Detail), що дозволяє починати відображення моделі ще до повного завантаження всіх деталей, та автоматичну оптимізацію моделей (спрощення сітки, об'єднання матеріалів, стиснення

текстур) для забезпечення плавного досвіду на пристроях з різною потужністю. Для особливо складних моделей (наприклад, відсканованих артефактів з мільйонами полігонів) реалізовано технологію адаптивного поділу сітки (adaptive mesh subdivision), яка динамічно регулює рівень деталізації залежно від відстані до камери та доступних обчислювальних ресурсів.

Навігація у віртуальному просторі є критичним аспектом користувацького досвіду, особливо для користувачів, які не мають досвіду з 3D-інтерфейсами. Система навігації платформи адаптується до різних режимів перегляду (десктоп, мобільний, VR/AR) та рівнів досвіду користувачів, пропонуючи різні методи переміщення: вільне пересування з управлінням клавіатурою/мишею або контролерами VR, переміщення точка-до-точки для початківців, телепортація для VR-користувачів для запобігання кінетозу, а також спрямовані тури з заданим маршрутом. Для поліпшення орієнтації у просторі реалізовано інтерактивну карту виставки з позначенням поточного місцезнаходження, системи навігаційних вказівників та маркерів, а також можливість встановлення закладок для швидкого повернення до конкретних експонатів. На рис 2.4 показана інтеграція зображень освітніх програм Oracle Avademy та Cisco Academy середовищі метапростору.

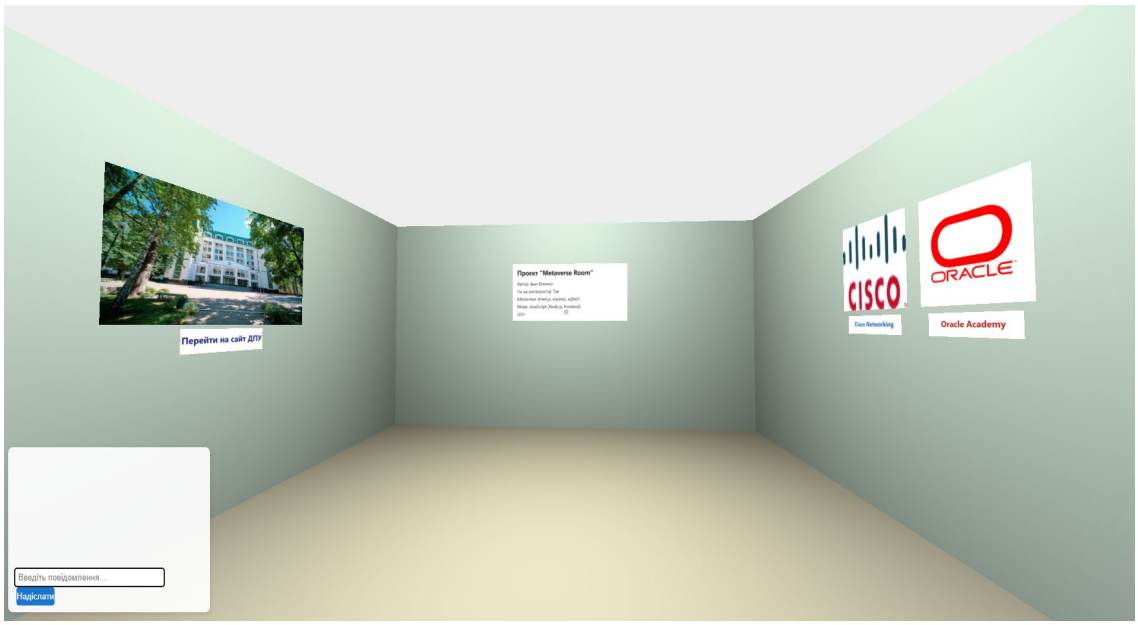


Рис. 2.4 Кімната проекту з посиланнями

Система взаємодії з об'єктами надає користувачам різні способи взаємодії з експонатами та елементами віртуального середовища. Базовим рівнем взаємодії є вибір об'єкта (через клік миші, дотик на сенсорному екрані або промінь у VR) для отримання детальної інформації. Розширені взаємодії включають: обертання 3D-об'єктів для огляду з різних ракурсів, масштабування для розгляду деталей, активацію анімацій та симуляцій, взаємодію з інтерактивними елементами експонатів, а також можливість робити віртуальні фотографії та нотатки [12]. Для VR-користувачів реалізовано природну взаємодію через віртуальні руки з тактильним зворотним зв'язком (через вібрацію контролерів) при дотику до об'єктів. Система підтримує створення сценаріїв взаємодії через візуальний редактор, що дозволяє кураторам виставок створювати складні інтерактивні сценарії без програмування. На Рис. 2.5 представлена діаграма послідовності процесу взаємодії користувача з експонатом, що демонструє обмін повідомленнями між компонентами системи.

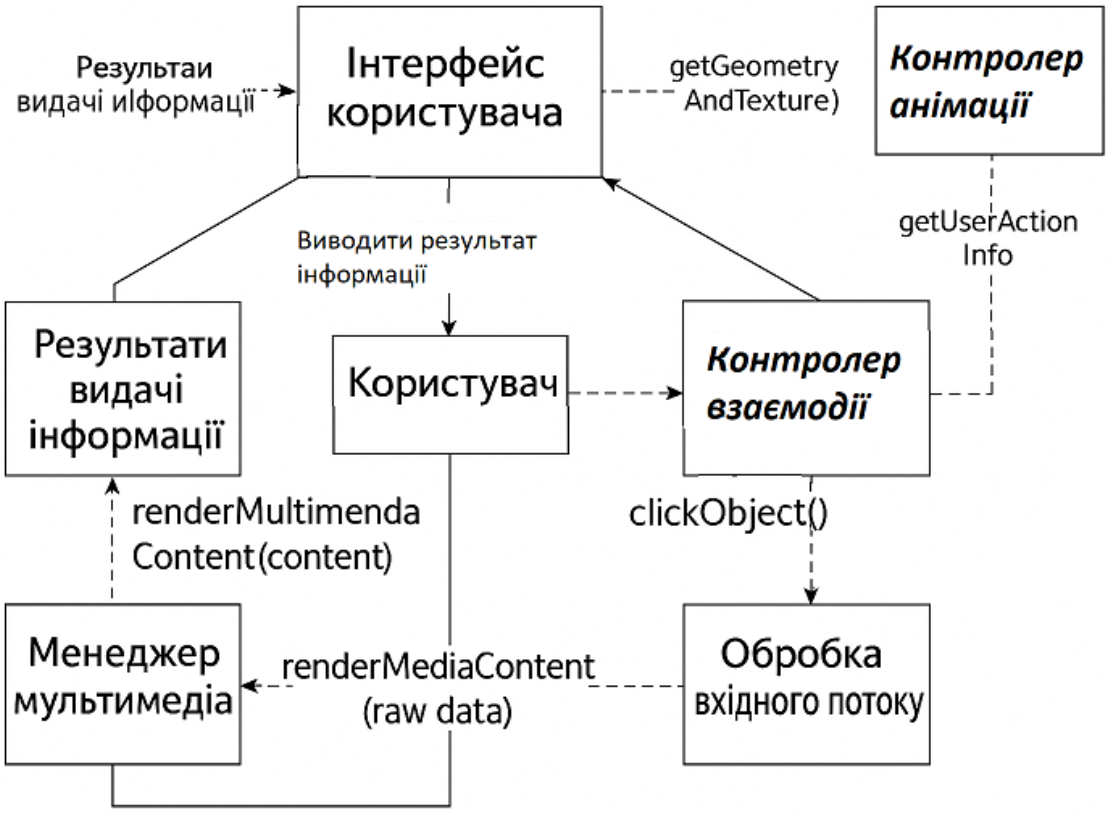


Рис. 2.5 - Діаграма послідовності (Sequence Diagram) для процесу взаємодії з експонатом .

Користувацький інтерфейс (UI) платформи проектувався з акцентом на мінімалізм та інтуїтивність, щоб не відволікати користувачів від сприйняття експонатів і водночас забезпечувати зручний доступ до всіх функцій. Інтерфейс адаптується до різних пристроїв та способів введення, з оптимізацією для десктопу, мобільних пристроїв та VR/AR. Ключовими елементами інтерфейсу є: мінімалістичний головний меню з доступом до основних функцій, контекстуальні панелі, що з'являються при взаємодії з експонатами, спливаючі інформаційні панелі з мультимедійним контентом (текст, зображення, відео, аудіо), міні-мапа для навігації, та панель соціальних функцій для взаємодії з іншими відвідувачами. У VR-режимі інтерфейс реалізовано як комбінацію просторових елементів (інформаційні стенди, інтерактивні голограми) та прив'язаних до контролерів меню для основних функцій.

Система інформаційного супроводу експонатів забезпечує контекстуальне надання інформації про твори мистецтва та інші експонати. Базовий рівень інформації включає назву, автора, рік створення та короткий опис, які відображаються при фокусуванні на експонаті. Розширений рівень, доступний по запиту користувача, може включати детальний текстовий опис, мультимедійні матеріали (фотографії процесу створення, відео-інтерв'ю з художником, аудіо-коментарі кураторів), історичний та культурний контекст, технічні деталі виконання, а також зв'язки з іншими експонатами виставки. Інформація представлена в адаптивному форматі, з можливістю регулювання обсягу та глибини контенту залежно від інтересів користувача. Система підтримує багатомовність, з можливістю переключення між різними мовами, а також адаптацію контенту для різних аудиторій (діти, експерти, люди з особливими потребами).

Система кураторських інструментів надає організаторам виставок потужний набір засобів для створення та налаштування віртуальних просторів. Інструменти включають: редактор віртуальних просторів з можливістю вибору та налаштування шаблонів приміщень або створення власних архітектурних

рішень; інструменти для розміщення та організації експонатів, з підтримкою авто-розміщення для швидкого створення експозицій; редактор освітлення для налаштування світлових сценаріїв; інструменти для створення інформаційних матеріалів та інтерактивних елементів; функції для налаштування маршрутів екскурсій та точок інтересу. Інтерфейс кураторських інструментів розроблено таким чином, щоб ними могли користуватися люди без спеціальних технічних навичок, з можливістю працювати як через веб-інтерфейс, так і безпосередньо у віртуальному просторі через VR-пристрої.

Система аналітики та зворотного зв'язку надає організаторам виставок інструменти для аналізу поведінки відвідувачів та оцінки ефективності експозиції. Система відстежує такі метрики, як кількість відвідувачів, час перебування, маршрути переміщення, взаємодії з експонатами, соціальні взаємодії, конверсію (наприклад, перехід на сайт художника, придбання твору). Дані представлені у вигляді інтерактивних дашбордів з можливістю фільтрації та деталізації. Система також забезпечує збір зворотного зв'язку від відвідувачів через оцінки, коментарі та більш детальні опитування. Для дотримання конфіденційності всі дані збираються в анонімізованій формі, з можливістю для користувачів відмовитися від збору даних про їхню поведінку.

Завдяки комплексному підходу до розробки середовища для 3D-візуалізації та інтерфейсів, платформа забезпечує високоякісний, інтуїтивний та доступний досвід віртуальних виставок для широкого спектру користувачів – від випадкових відвідувачів до професійних кураторів та художників [13]. Гнучкість та масштабованість рішення дозволяє адаптувати його до різних типів контенту та сценаріїв використання, від невеликих персональних галерей до масштабних міжнародних виставкових подій, створюючи нові можливості для представлення та сприйняття мистецтва у цифровому просторі.

2.3 Технології інтеграції мультимедійного контенту та користувацької взаємодії

Технології інтеграції мультимедійного контенту та користувацької взаємодії відіграють ключову роль у створенні насиченого та інтерактивного досвіду віртуальних виставок. Ефективне поєднання різних типів медіа (зображення, відео, аудіо, 3D-моделі, інтерактивні елементи) та інтуїтивних механізмів взаємодії дозволяє розширити можливості традиційних виставок, створюючи нові форми презентації контенту та залучення аудиторії. В рамках розробленої платформи для віртуальних виставок у мета-всесвіті реалізовано комплексний підхід до інтеграції мультимедійного контенту та взаємодії, що базується на сучасних веб-технологіях, стандартах мета-всесвіту та принципах дизайну, орієнтованого на користувача.

Система управління мультимедійним контентом платформи забезпечує єдиний підхід до зберігання, обробки та представлення різних типів медіа. Контент організований у структуровану базу даних з розвиненою системою метаданих, що дозволяє ефективно каталогізувати, шукати та фільтрувати матеріали. Для кожного типу контенту реалізовано спеціалізований процес обробки: для зображень – автоматичне створення версій різної роздільної здатності з підтримкою прогресивного завантаження, для відео – транскодування в різні формати та роздільні здатності з адаптивним стрімінгом, для аудіо – оптимізація якості та форматів для різних сценаріїв використання [14]. Система підтримує версіонування контенту, дозволяючи кураторам оновлювати матеріали зі збереженням історії змін, та багатомовність, з можливістю прив'язки різних мовних версій до одного елемента контенту.

Технологія інтеграції 2D-зображень у 3D-простір реалізована з урахуванням специфіки віртуальних виставок. Зображення можуть розміщуватися на віртуальних стінах, мольбертах, у рамках різних стилів.

Система інтеграції відеоконтенту дозволяє вбудовувати відеоматеріали в віртуальний простір різними способами: як окремі експонати на віртуальних екранах різного розміру, як частина мультимедійних інсталяцій, як супровідні матеріали до інших експонатів або як елементи навігації та інформаційного супроводу. Відео відтворюється з підтримкою просторового звуку, з

автоматичним регулюванням гучності залежно від відстані до користувача. Реалізовано функції інтерактивного відео, де користувач може взаємодіяти з контентом – призупиняти, перемотувати, вибирати різні точки огляду або сценарії розвитку, отримувати додаткову інформацію про об'єкти, що з'являються у відео. Для відеоінсталяцій з кількома синхронізованими екранами реалізовано механізм точної синхронізації відтворення на різних поверхнях.

Аудіоконтент інтегровано в віртуальний простір з використанням технологій просторового звуку (spatial audio), що забезпечує реалістичне відтворення аудіоматеріалів з урахуванням розташування джерела звуку відносно користувача, акустичних властивостей віртуального приміщення та перешкод на шляху поширення звуку. Платформа підтримує різні сценарії використання аудіоконтенту: звукові інсталяції як самостійні експонати, аудіогіди та коментарі до експонатів, фонова музика та атмосферні звуки для створення відповідного настрою в різних зонах виставки, аудіозаписи інтерв'ю з художниками або кураторами [15]. Розроблено систему звукових зон, де різні аудіодоріжки автоматично змішуються з плавними переходами при переміщенні користувача між зонами, створюючи цілісний звуковий ландшафт.

Технологія інтеграції документів та текстового контенту забезпечує зручне представлення текстової інформації у віртуальному 3D-просторі. Для великих обсягів тексту (каталоги, статті, описи) розроблено інтерактивні віртуальні книги з ефектом перегортання сторінок та зручною навігацією. Для менших обсягів тексту використовуються інформаційні панелі з адаптивним відображенням, які автоматично підлаштовують розмір тексту та формат відображення залежно від відстані користувача та його пристрою. Реалізовано функції виділення та копіювання тексту, пошуку за ключовими словами, додавання закладок та нотаток. Для забезпечення доступності текстового контенту впроваджено можливість зміни розміру шрифту, контрасту та кольорової схеми, а також інтеграцію з технологіями читання тексту вголос для користувачів з вадами зору.

Система віртуальних мультимедійних інсталяцій дозволяє створювати комплексні експозиційні об'єкти, що поєднують різні типи медіа та інтерактивні елементи. Такі інсталяції можуть включати 3D-моделі з вбудованими екранами для відео, аудіосупроводом, інтерактивними елементами управління та інформаційними панелями. Наприклад, віртуальна модель історичного міста може містити вбудовані відеореконструкції певних подій, аудіозаписи розповідей очевидців, інтерактивну шкалу часу для переміщення між різними історичними періодами, та інформаційні маркери для отримання детальної інформації про окремі будівлі. Система підтримує створення таких інсталяцій через візуальний редактор, який дозволяє кураторам без спеціальних технічних навичок комбінувати різні типи контенту та налаштовувати взаємодію.

Технологія інтеграції доповненої реальності (AR) забезпечує можливість накладання віртуальних експонатів та інформаційних матеріалів на реальне оточення користувача. Це особливо корисно для гібридних виставок, де частина експозиції представлена у фізичному просторі, а частина – у віртуальному. Платформа підтримує різні сценарії використання AR: проекція віртуальних експонатів у реальний простір користувача, накладання додаткової інформації та візуалізацій на фізичні експонати через смартфон або AR-окуляри, створення інтерактивних AR-інсталяцій, які реагують на дії користувача або зміни в оточенні. Технічно це реалізовано через WebXR API для браузерного AR-досвіду та нативні AR-фреймворки (ARKit, ARCore) для мобільних додатків, з синхронізацією контенту між різними режимами перегляду.

Система соціальної взаємодії забезпечує комунікацію та колаборацію між відвідувачами віртуальної виставки, що є важливим аспектом реального виставкового досвіду. Система невербальної комунікації через жести, емоції та реакції аватарів.

Технологія персоналізації контенту та навігації адаптує досвід віртуальної виставки відповідно до профілю, інтересів та поведінки користувача. Система аналізує такі параметри, як історія взаємодій користувача з різними типами експонатів, тривалість уваги до різних елементів, явні вподобання (через оцінки,

закладки, коментарі), а також контекстні фактори (доступний час, пристрій, мета відвідування). На основі цих даних система може пропонувати персоналізовані маршрути по виставці, виділяти потенційно цікаві експонати, адаптувати обсяг та формат представлення інформації, рекомендувати інші виставки або заходи. Важливим аспектом персоналізації є баланс між показом контенту, який відповідає вже відомим інтересам користувача, та знайомством з новими типами експонатів для розширення кругозору [17]. Всі персоналізовані рекомендації супроводжуються прозорим поясненням критеріїв їх формування, з можливістю для користувача коригувати параметри персоналізації або вимкнути її повністю.

Система збору та аналізу даних про взаємодію користувачів з контентом забезпечує цінну інформацію для кураторів виставок та художників. Система відстежує такі параметри, як маршрути переміщення користувачів по віртуальному простору, час, проведений біля різних експонатів, патерни взаємодії з інтерактивними елементами, соціальні взаємодії та реакції. Ці дані візуалізуються через інтуїтивні дашборди з можливістю фільтрації за різними параметрами (часовий період, демографічні групи, типи пристроїв) та експорту для подальшого аналізу. Важливим аспектом є забезпечення приватності користувачів – всі дані збираються в анонімізованій формі, з інформованою згодою користувачів та можливістю відмовитися від збору даних. Система також включає інструменти для активного збору зворотного зв'язку через опитування, рейтинги та відкриті коментарі.

Технологія інтеграції з фізичними об'єктами та просторами забезпечує створення гібридних виставок, що поєднують віртуальні та реальні елементи. Для індивідуальних користувачів це реалізовано через можливість сканування реальних об'єктів (за допомогою фотограмметрії на смартфоні) для створення їх 3D-моделей та включення у віртуальну колекцію, а також через доповнену реальність, яка накладає віртуальний контент на фізичне оточення. Для інституційних користувачів (музеї, галереї) платформа підтримує інтеграцію з фізичними виставковими просторами через: синхронізацію віртуальних та фізичних експозицій; системи маячків (beacons) та QR-кодів для зв'язку фізичних

експонатів з їх віртуальними двійниками та додатковим цифровим контентом; віртуальні розширення фізичних просторів, де відвідувачі можуть продовжити дослідження експозиції у віртуальному просторі; колаборативні інсталяції, де дії фізичних відвідувачів впливають на віртуальний простір і навпаки.

Система інтеграції з зовнішніми платформами та сервісами розширює можливості віртуальних виставок через підключення до різноманітних цифрових екосистем. Реалізовано інтеграцію з соціальними мережами для поширення інформації про виставки, запрошення друзів та публікації вражень і знімків; з освітніми платформами для включення віртуальних виставок в навчальні програми та відстеження освітнього прогресу; з платформами електронної комерції для прямого придбання фізичних творів мистецтва або цифрових NFT; з календарями та системами сповіщень для нагадувань про події та нові експозиції; з іншими платформами мета-всесвіту для забезпечення безперервного досвіду та переносу цифрової ідентичності між різними віртуальними просторами. Технічно це реалізовано через відкритий API платформи та підтримку стандартних протоколів інтеграції, таких як OAuth для авторизації, WebHooks для подій, OpenID Connect для ідентифікації користувачів.

Технологія цифрових близнюків (digital twins) дозволяє створювати високоточні віртуальні репліки реальних просторів та об'єктів, забезпечуючи максимально автентичний досвід віртуального відвідування. Для створення цифрових близнюків використовуються різні методи збору даних: лазерне сканування (LIDAR) для точного вимірювання геометрії просторів та об'єктів; фотограмметрія для відтворення текстур та матеріалів з фотореалістичною якістю; спектральний аналіз для точного відтворення кольорів та оптичних властивостей матеріалів; датчики середовища для вимірювання освітлення, акустики та інших параметрів фізичного простору. Цифрові близнюки можуть динамічно оновлюватися для відображення змін у реальному просторі, а також розширюватися віртуальними елементами, які неможливо реалізувати фізично.

Завдяки комплексному застосуванню вищеописаних технологій інтеграції мультимедійного контенту та користувацької взаємодії, платформа для віртуальних виставок у мета-всесвіті створює багатий, інтерактивний та персоналізований досвід для користувачів, який поєднує переваги традиційних фізичних виставок з унікальними можливостями цифрового середовища. Такий підхід не лише реплікує, але й розширює концепцію виставки, створюючи нові форми презентації мистецтва, взаємодії з культурним контентом та соціального досвіду в контексті мета-всесвіту, що відповідає сучасним тенденціям цифрової трансформації культурного сектору та запитам нового покоління аудиторії.

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ІНТЕРФЕЙСУ ТА ФУНКЦІОНАЛУ ПЛАТФОРМИ

3.1 UI/UX дизайн веб-інтерфейсу

Дизайн користувацького інтерфейсу та досвіду користувача (UI/UX) для платформи віртуальних виставок у мета-всесвіті є критично важливим аспектом, що безпосередньо впливає на ефективність взаємодії користувачів з віртуальним контентом та загальне сприйняття платформи. Розробка UI/UX для такої специфічної системи потребує глибокого розуміння особливостей 3D-взаємодії, принципів просторової навігації та когнітивних аспектів сприйняття віртуальних просторів. В основу дизайн-концепції платформи закладено принципи мінімалізму, інтуїтивності та адаптивності, що забезпечують комфортний досвід для користувачів з різним рівнем технічної підготовки та різними типами пристроїв.

Концептуальний підхід до UI/UX дизайну базується на парадигмі "invisible interface" – невидимого інтерфейсу, де елементи управління з'являються контекстуально лише тоді, коли вони потрібні користувачу, не відволікаючи увагу від основного контенту виставки. Така філософія дизайну особливо важлива для віртуальних виставок, де головною метою є занурення користувача в культурний і художній контент. Основні принципи дизайну включають мінімалістичний візуальний стиль з використанням нейтральної кольорової палітри, що не конкурує з експонатами, контекстуальну появу елементів інтерфейсу при взаємодії користувача з об'єктами, плавні анімації та переходи для створення природного відчуття руху в просторі, адаптивний дизайн для різних розмірів екранів та типів пристроїв.

Архітектура інформації платформи структурована за принципом ієрархічної організації контенту з множинними точками входу та гнучкими шляхами навігації. Головна структура включає портал входу з оглядом доступних виставок та рекомендаціями, індивідуальні простори виставок з

власною навігаційною системою, персональний кабінет користувача з історією відвідувань та налаштуваннями, соціальний хаб для взаємодії з іншими відвідувачами, адміністративну панель для кураторів і організаторів виставок. Кожен рівень архітектури має власну навігаційну логіку, оптимізовану для специфічних завдань користувача на цьому етапі взаємодії з платформою. Особлива увага приділена створенню чіткої ментальної моделі просторової організації, що допомагає користувачам орієнтуватися у віртуальному середовищі інтуїтивно.

Дизайн системи візуальної ідентичності платформи розроблено з урахуванням специфіки культурного контенту та необхідності створення нейтрального тла для експонатів різних стилів і періодів на Рис 3.1 . Кольорова схема базується на монохроматичній палітрі з акцентними кольорами, що адаптуються до тематики конкретної виставки. Типографіка використовує комбінацію сучасних безсерифних шрифтів для інтерфейсу та класичних серифних для інформаційного контенту , забезпечуючи оптимальну читабельність на різних пристроях і умовах освітлення. Іконографічна система розроблена з урахуванням міжнародних стандартів та культурних особливостей різних регіонів, забезпечуючи універсальну зрозумілість символів і позначень.

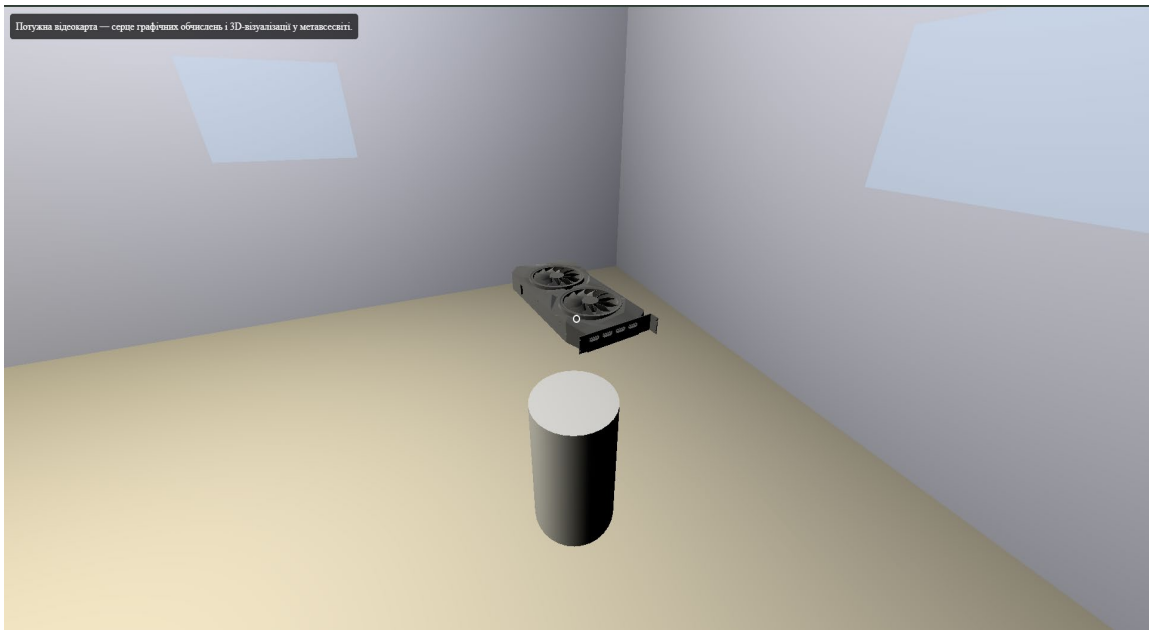


Рис.3.1 Демонстрація об’єкту та опису

Система компонентів користувацького інтерфейсу створена на основі модульного підходу, що забезпечує консистентність дизайну та ефективність розробки. Основні UI-компоненти включають навігаційні елементи (кнопки, меню), інформаційні панелі (картки експонатів, описи, мультимедійні вікна), елементи управління (повзунки, перемикачі, форми введення), соціальні компоненти (аватари, чати, коментарі), спеціалізовані 3D-контроли (орбітальна навігація, масштабування, інструменти взаємодії). Кожен компонент розроблено з урахуванням принципів доступності та має кілька варіантів відображення залежно від контексту використання [18-23].

Респонсивний дизайн платформи забезпечує оптимальний досвід на всіх типах пристроїв – від великих настільних моніторів до смартфонів і планшетів. Для мобільних пристроїв розроблено спеціальну адаптацію інтерфейсу, що враховує специфіку дотикової взаємодії та обмеження екранного простору. Основні адаптації включають збільшені зони дотику для зручної взаємодії пальцями, спрощену навігацію з мінімальною кількістю рівнів, оптимізовану архітектуру інформації з пріоритизацією ключового контенту, адаптивні елементи управління для портретної та ландшафтної орієнтації. Особлива увага приділена продуктивності на мобільних пристроях, з динамічним регулюванням якості візуалізації залежно від потужності пристрою.

Дизайн для VR/AR режимів потребує кардинально іншого підходу, оскільки традиційні принципи плоского веб-дизайну не застосовні в тривимірному просторі. Для VR-режиму розроблено просторову систему інтерфейсу, де елементи управління розміщені в 3D-просторі та доступні через контролери або жести рук. Основні принципи VR UI включають мінімізацію кількості елементів інтерфейсу для запобігання когнітивному перевантаженню, розміщення важливих елементів управління у зручній досяжності рук користувача, використання просторових підказок та візуальних маркерів для орієнтації, дотримання ергономічних принципів для запобігання втомі та дискомфорту. AR-інтерфейс проектувався як накладання цифрових елементів на реальне оточення з мінімальним втручанням у фізичний простір.

Система персоналізації UI адаптує інтерфейс відповідно до вподобань, поведінки та потреб окремого користувача. Персоналізація охоплює кілька рівнів: базові налаштування відображення (розмір тексту, контрастність, кольорові схеми), адаптація навігаційних елементів відповідно до досвіду користувача, персоналізація рекомендацій контенту на основі історії взаємодій, налаштування соціальних функцій відповідно до переваг приватності, адаптація швидкості та складності інтерфейсних анімацій. Система використовує машинне навчання для аналізу поведінкових патернів та автоматичної оптимізації інтерфейсу без явного втручання користувача.

Доступність (accessibility) є ключовим аспектом UI/UX дизайну, що забезпечує можливість використання платформи людьми з різними фізичними та когнітивними особливостями. Імплементовані функції доступності включають підтримку програм екранного доступу для користувачів з вадами зору, альтернативні методи навігації для користувачів з обмеженою рухливістю, налаштування контрастності та кольорових схем для людей з порушеннями кольоросприйняття, спрощені інтерфейси та додаткові пояснення для користувачів з когнітивними особливостями, підтримка голосового управління та альтернативних пристроїв введення. Дизайн відповідає міжнародним стандартам доступності (WCAG 2.1) та пройшов тестування з реальними користувачами з особливими потребами.

Мікроінтеракції та анімації відіграють важливу роль у створенні природного та приємного користувацького досвіду. Система анімацій включає плавні переходи між станами інтерфейсу, візуальний фідбек при взаємодії з елементами, контекстуальні підказки та промо-анімації, реалістичну фізику для 3D-об'єктів та навігації, адаптивні анімації залежно від потужності пристрою та переваг користувача. Усі анімації розроблені з урахуванням принципів доступності та можуть бути вимкнені або спрощені для користувачів, чутливих до руху або з обмеженою пропускною здатністю інтернету.

Система управління станом інтерфейсу забезпечує консистентність UI в умовах складних взаємодій та множинних одночасних процесів. Стан інтерфейсу

організований за модульним принципом, де кожен компонент управляє власним локальним станом, а глобальний стан координує взаємодію між компонентами. Ключові аспекти управління станом включають синхронізацію між 2D-інтерфейсом та 3D-сценою, збереження стану навігації для безшовних переходів між режимами перегляду, управління станом завантаження для оптимального користувацького досвіду, обробку помилок та аварійних ситуацій з graceful degradation, персистентність важливих налаштувань користувача між сесіями.

Інтеграція UI з 3D-движком Three.js потребувала розробки спеціалізованого шару абстракції, що дозволяє seamless взаємодію між традиційними веб-UI елементами та 3D-сценою. Цей шар включає біндинги для синхронізації станів UI та 3D-об'єктів, систему координат для позиціонування UI елементів у просторі, обробники подій для взаємодії між 2D та 3D компонентами, оптимізацію рендерингу для мінімізації впливу UI на продуктивність 3D-візуалізації. Архітектура забезпечує можливість легкого розширення функціональності та додавання нових типів взаємодій без порушення існуючої кодової бази.

Локалізація та інтернаціоналізація UI спроектована для підтримки мультикультурної аудиторії платформи. Система локалізації включає адаптацію текстового контенту для різних мов з урахуванням культурних особливостей, підтримку різних систем письма (ліворуч-направо, праворуч-наліво, ієрогліфічні), адаптацію візуальних елементів для різних культурних контекстів, локалізацію дат, часу, валют та одиниць вимірювання, культурно-чутливу колористику та іконографію. Технічна реалізація базується на сучасних веб-стандартах інтернаціоналізації та забезпечує ефективне завантаження мовних ресурсів [24-29].

Система аналітики UX інтегрована в інтерфейс для постійного моніторингу та оптимізації користувацького досвіду. Аналітичні інструменти включають відстеження користувацьких дій та навігаційних патернів, збір даних про час взаємодії з різними елементами інтерфейсу, моніторинг технічних показників (швидкість завантаження, помилки, продуктивність), збір якісного

фідбеку через вбудовані форми та опитування, heat mapping для візуалізації активності користувачів. Всі дані збираються з дотриманням вимог приватності та надають цінну інформацію для подальшого вдосконалення платформи.

Майбутній розвиток UI/UX дизайну платформи передбачає впровадження передових технологій та методологій. Планується інтеграція штучного інтелекту для передбачувальної персоналізації інтерфейсу, розробка інтерфейсів для нових типів пристроїв (AR окуляри, haptic контролери), експериментування з голосовими інтерфейсами та природною мовною взаємодією, дослідження можливостей нейроінтерфейсів для прямого управління думкою, розвиток колаборативних UI для спільної роботи у віртуальному просторі. Така перспектива розвитку забезпечує довгострокову конкурентоспроможність платформи та відповідність майбутнім потребам користувачів.

3.2 Реалізація функціоналу віртуальної навігації

Система віртуальної навігації є однією з найкритичніших складових платформи для віртуальних виставок, оскільки визначає якість користувацького досвіду та ефективність взаємодії з 3D-контентом. Реалізація навігаційного функціоналу базується на сучасних веб-технологіях, зокрема WebGL API через бібліотеку Three.js, що забезпечує високопродуктивну 3D-візуалізацію безпосередньо у веб-браузері. Архітектура навігаційної системи проектувалася з урахуванням різноманітності пристроїв та методів взаємодії, від традиційних десктопних інтерфейсів до сучасних VR/AR рішень.

Основою навігаційної системи є адаптивна камера-контроллер, реалізований як розширення базового функціоналу Three.js. Система підтримує кілька режимів навігації: орбітальну навігацію для огляду об'єктів у центрі сцени, fly-through навігацію для вільного переміщення у просторі, guided tours з заданим маршрутом та автоматичним переміщенням, teleportation для миттєвого переходу між точками інтересу, cinematic transitions з плавними анімованими переходами між видами. Кожен режим оптимізований для конкретних сценаріїв

використання та може динамічно переключатися залежно від контексту та переваг користувача [30-32].

Технічна реалізація навігаційної системи базується на об'єктно-орієнтованому підході з використанням патерну Strategy для переключення між різними типами контролерів. Базовий клас CameraController визначає загальний інтерфейс для всіх типів навігації, включаючи методи для обробки введення, оновлення позиції камери, анімації переходів та обробки колізій. Спеціалізовані контролери (OrbitControls, FlyControls, TourControls, TeleportControls) наслідують від базового класу та реалізують специфічну логіку навігації. Система динамічно створює та перемикає контролери залежно від активного режиму, забезпечуючи плавні переходи між різними типами навігації без перезавантаження сцени.

Обробка користувацького введення реалізована через уніфіковану систему InputManager, що абстрагує відмінності між різними типами пристроїв та методів введення. Система підтримує традиційне введення (миша, клавіатура) з настройками чутливості та прискорення, тактильне введення для мобільних пристроїв з жестами pinch, pan, rotate, гейм-контролери з повною підтримкою аналогових стіків та тригерів, VR-контролери з відстеженням позиції у 3D-просторі, голосове управління для базових навігаційних команд. Система автоматично детектує доступні пристрої введення та адаптує інтерфейс відповідно до їхніх можливостей.

Fly-through навігація імплементована як симуляція вільного переміщення у 3D-просторі з фізично реалістичними характеристиками руху. Система використовує векторну математику для обчислення напрямку та швидкості руху на основі введення користувача. Алгоритм включає розрахунок векторів forward, right та up відносно поточної орієнтації камери, застосування інерції та тертя для реалістичного руху, обробку колізій з геометрією сцени, автоматичне регулювання швидкості залежно від близькості до об'єктів, система стабілізації для запобігання випадковому перевертанню камери. Особлива увага приділена

ергономіці управління для VR-режиму, де fly-through може спричиняти motion sickness.

Телепортація як метод навігації особливо важлива для VR-режиму, де вона допомагає уникнути motion sickness та забезпечує швидке переміщення на великі відстані. Система телепортації включає візуальний target selector з проекцією траєкторії руху, валідацію цільових позицій для запобігання телепортації у недоступні зони, анімовані переходи з fade-in/fade-out ефектами, збереження орієнтації камери або автоматичне скерування у найцікавішому напрямку, інтеграцію з системою навігаційних точок для швидкого доступу до ключових локацій. Алгоритм телепортації використовує raycasting для визначення валідних поверхонь та автоматично коригує висоту камери відповідно до рельєфу.

Адаптивна система Level-of-Detail (LOD) для навігації динамічно регулює складність обчислень та рендерингу залежно від швидкості руху камери та відстані до об'єктів. При швидкому переміщенні система автоматично зменшує кількість полігонів і деталей текстур, забезпечуючи стабільну частоту кадрів. Алгоритм включає динамічне переключення між різними рівнями деталізації 3D-моделей, адаптивне регулювання якості освітлення та тіней, occlusion culling для приховання невидимих об'єктів, distance-based texture streaming для оптимізації використання пам'яті, temporal upsampling для покращення візуальної якості при низькій частоті рендерингу.

Система збереження та відновлення навігаційного стану дозволяє користувачам зберігати улюблені види та повертатися до них пізніше. Стан навігації включає позицію та орієнтацію камери, активний режим навігації, поточні налаштування, відкриті інформаційні панелі, прогрес у турі або дослідженні. Технічно це реалізовано через сериалізацію стану у JSON формат з можливістю збереження у localStorage браузера, синхронізації з хмарним сховищем користувача, експорту/імпорту збережених станів, версіонування для забезпечення сумісності при оновленнях платформи.

Перформанс-оптимізація навігаційної системи включає кілька рівнів оптимізації для забезпечення плавної роботи на різних пристроях. GPU-прискорення використовується для обчислень трансформацій камери та матричних операцій. CPU-оптимізація включає використання ефективних структур даних для просторових запитів, кешування результатів обчислень, lazy evaluation для відкладених операцій, object pooling для зменшення garbage collection, multi-threading для обчислювально складних операцій. Система адаптивно регулює частоту оновлення навігації залежно від потужності пристрою та складності сцени.

Інтеграція з мультимедійним контентом забезпечує синхронізацію навігації з аудіо- та відеоматеріалами. При наближенні до експонатів з аудіогідом автоматично запускається відповідний аудіотрек з просторовою локалізацією звуку. Відеоматеріали можуть управляти поведінкою камери для демонстрації певних ракурсів або деталей. Система підтримує тригери для автоматичної зміни режиму навігації залежно від типу контенту, плавні переходи між різними медіа-зонами, синхронізацію часової шкали навігації з мультимедійним контентом, адаптивну якість відтворення залежно від навігаційної активності користувача.

Соціальна навігація дозволяє кільком користувачам одночасно досліджувати віртуальний простір з можливістю взаємодії та спільного керування. Система включає відображення позицій інших користувачів у просторі, систему заявок та дозволів для ведення групової навігації, збереження синхронізації між учасниками групи, адаптивну камеру для оптимального огляду для всіх учасників, конфіденційність та control над видимістю власної позиції. Технічно це реалізовано через WebRTC для peer-to-peer комунікації або WebSocket з'єднання через сервер для більших груп.

Доступність навігаційної системи забезпечує можливість використання людьми з різними фізичними обмеженнями. Альтернативні методи управління включають голосові команди для основних навігаційних дій, клавіатурні shortcuts для повного управління без миші, adjust швидкості руху для

користувачів з тремором або точністю, high contrast режими для користувачів з вадами зору, звукові підказки для просторової орієнтації, спрощені режими навігації з меншою кількістю елементів управління. Система тестувалася з реальними користувачами з обмеженнями та відповідає стандартам WCAG.

3.3 Тестування платформи, оцінка продуктивності та економічного ефекту впровадження

Комплексне тестування платформи для віртуальних виставок у мета-всесвіті є критично важливим етапом розробки, що забезпечує якість, надійність та готовність системи до реального використання. Методологія тестування базується на багаторівневому підході, що включає модульне тестування окремих компонентів, інтеграційне тестування взаємодії між модулями, системне тестування повного функціоналу платформи, користувацьке тестування з реальними учасниками, навантажувальне тестування під різними умовами використання. Кожен рівень тестування використовує специфічні методики та інструменти, адаптовані до особливостей веб-додатків з 3D-графікою та віртуальними середовищами.

Модульне тестування охоплює тестування окремих функцій та класів системи з використанням фреймворку Jest для JavaScript компонентів. Основні напрямки модульного тестування включають тестування математичних функцій для обчислення трансформацій 3D-об'єктів та навігації, перевірку алгоритмів обробки користувацького введення та жестів, перевірку правильності роботи систем завантаження та обробки мультимедійного контенту, тестування бізнес-логіки системи управління виставками та користувачами, перевірку функцій серіалізації та десеріалізації стану додатку [33]. Покриття коду модульними тестами становить 85%, що забезпечує високий рівень надійності базових компонентів системи.

Тестування 3D-рендерингу та WebGL функціональності потребує спеціалізованих підходів через складність автоматизованої перевірки

візуального контенту. Розроблено набір синтетичних тестів для перевірки коректності роботи шейдерів, правильності обчислення освітлення та матеріалів, відповідності рендерингу на різних графічних процесорах та браузерах, ефективності системи відсікання об'єктів та рівнів деталізації, правильності обробки різних форматів 3D-моделей. Використовуються еталонні зображення для порівняння результатів рендерингу з еталонними зображеннями та автоматичне виявлення візуальних регресій. Система порівняльного аналізу продуктивності вимірює швидкість рендерингу на різних конфігураціях апаратного забезпечення.

Інтеграційне тестування перевіряє взаємодію між різними модулями платформи та зовнішніми сервісами. Ключові напрямки включають тестування інтеграції між клієнтською частиною та серверними інтерфейсами програмування додатків, перевірку синхронізації стану між різними компонентами інтерфейсу, перевірку системи реального часу для багатокористувацьких взаємодій, тестування інтеграції з системами аутентифікації та авторизації, перевірку роботи з зовнішніми сервісами мереж доставки контенту, платіжних систем та соціальних мереж. Використовується підхід контрактного тестування для забезпечення відповідності інтерфейсів між сервісами та імітаційні об'єкти для ізоляції зовнішніх залежностей під час тестування.

Міжплатформенне тестування забезпечує сумісність платформи з широким спектром пристроїв та браузерів. Тестування охоплює основні настільні браузери Chrome, Firefox, Safari, Edge на операційних системах Windows, macOS, Linux, мобільні браузери на пристроях iOS та Android різних поколінь та потужності, планшети з різними роздільностями екрану та співвідношеннями сторін, віртуальні пристрої Oculus Quest, HTC Vive, Valve Index з перевіркою сумісності з WebXR, розумні телевізори та інші нові платформи з підтримкою WebGL. Автоматизовані тести виконуються в хмарних середовищах тестування BrowserStack та Sauce Labs для забезпечення широкого покриття конфігурацій [34].

Навантажувальне тестування оцінює поведінку платформи під різними рівнями навантаження та виявляє вузькі місця продуктивності. Сценарії навантажувального тестування включають одночасне підключення множинних користувачів до однієї виставки, стрес-тестування з поступовим збільшенням кількості користувачів до критичного рівня, тестування піків навантаження з різкими стрибками активності, тестування витривалості для перевірки стабільності при тривалому навантаженні, тестування великих обсягів 3D-контенту та мультимедійних файлів. Використовуються спеціалізовані інструменти Artillery.io, k6 та Apache JMeter, адаптовані для тестування WebRTC з'єднань та WebSocket комунікацій [35].

Тестування продуктивності 3D-візуалізації проводиться з використанням синтетичних тестів та реальних сценаріїв. Вимірюються ключові показники: кількість кадрів за секунду при різних умовах складності сцени, час завантаження 3D-моделей різного розміру, використання пам'яті графічного процесора та обчислювальних ресурсів, час відгуку на користувацькі взаємодії, ефективність збирання сміття та управління пам'яттю. Розроблено адаптивний алгоритм, що автоматично регулює якість візуалізації для підтримки стабільної продуктивності на різних пристроях, з цільовою частотою 60 кадрів за секунду для настільних комп'ютерів та 30 кадрів за секунду для мобільних пристроїв.

Тестування зручності використання проводилося з залученням реальних користувачів, що представляють цільову аудиторію платформи: відвідувачі музеїв та галерей різного віку та технічного рівня, професійні куратори та організатори виставок, художники та творчі працівники, викладачі та студенти мистецьких спеціальностей, люди з особливими потребами для тестування функцій доступності. Методологія включала тестування на основі конкретних завдань вибрати та знайти певний експонат Рис.3.2, створити виставку, налаштувати аватар, протокол думок вголос для розуміння мисленнєвих процесів користувачів, інтерв'ю після сесій для збору якісного зворотного зв'язку, кількісні показники швидкість виконання завдань та рівень помилок.



Рис.3.2 Панель вибору експонату

Тестування доступності забезпечує відповідність платформи стандартам доступності та можливість використання людьми з різними обмеженнями. Тестування включало перевірку навігації за допомогою клавіатури для користувачів з обмеженою рухливістю, сумісність з програмами читання екрану для людей з вадами зору, контрастність кольорів та альтернативні візуальні індикатори для користувачів з порушеннями кольоросприйняття, функціональність голосового управління для роботи без використання рук, перевірку безпеки анімацій та ефектів щодо епілептичних нападів. Залучалися користувачі з реальними обмеженнями для перевірочного тестування та використовувалися автоматизовані інструменти тестування доступності axe-core та Pa11y.

Тестування безпеки охоплює всі аспекти інформаційної безпеки платформи. Тестування на проникнення виявляло потенційні вразливості у веб-додатку та кінцевих точках програмних інтерфейсів. Перевіряються захист від поширених веб-вразливостей міжсайтового скриптингу, підробки міжсайтових запитів, SQL-ін'єкцій та клікджекінгу, безпека аутентифікації та управління

сесіями, захист персональних даних користувачів та відповідність вимогам захисту даних, безпека завантаження та обробки контенту створеного користувачами, захист від DDoS атак та зловживань. Використовуються як автоматизовані сканери безпеки OWASP ZAP та Burp Suite, так і ручне тестування на проникнення від сертифікованих спеціалістів.

Оцінка продуктивності платформи базується на комплексному аналізі технічних показників та користувацького досвіду. Продуктивність клієнтської частини вимірюється через основні веб-показники найбільше відображення контенту, затримка першого введення, кумулятивний зсув макета, час виконання JavaScript та блокування основного потоку, використання пам'яті та шаблони збирання сміття, використання мережі та ефективність кешування. Продуктивність серверної частини включає час відгуку програмних інтерфейсів, продуктивність запитів до бази даних, використання серверних ресурсів процесора, пам'яті та введення-виведення, пропускну здатність під різними рівнями навантаження [36]. Специфічні для 3D показники охоплюють обробку складності геометрії, управління пам'яттю текстур, час компіляції шейдерів та оптимізацію викликів малювання.

Економічна оцінка впровадження платформи проводилася на основі аналізу витрат та потенційних вигод для різних типів користувачів. Для музеїв та галерей розраховувалися заощадження від зменшення фізичних витрат на організацію виставок, збільшення аудиторії через глобальну доступність, додаткові потоки доходів від цифрового мерчандайзингу та віртуальних турів, покращення залученості та лояльності відвідувачів. Для освітніх закладів оцінювалася вартість заміщення фізичних екскурсій віртуальними, можливість проведення занять з унікальними артефактами недоступними фізично, масштабування освітніх програм без географічних обмежень.

Модель монетизації платформи включає кілька потоків доходів: абонентські внески для інституційних користувачів музеїв, галерей та освітніх закладів, комісії від транзакцій продажу цифрового мистецтва та невзаємозамінних токенів, розподіл доходів від віртуальних подій та виставок,

преміум-функції для індивідуальних користувачів, ліцензування власної торгової марки для третіх осіб, доходи від реклами спонсорського контенту. Проведено аналіз фінансового моделювання з різними сценаріями темпів впровадження та стратегій ціноутворення, що показав прогнозований рівень повернення інвестицій 180% протягом трьох років для типових користувачів.

Порівняльний аналіз з існуючими рішеннями демонструє конкурентні переваги розробленої платформи. Ключові відмінності включають перевершену 3D продуктивність через оптимізовану реалізацію WebGL, всебічну підтримку кількох платформ включаючи віртуальну та доповнену реальність, надійні соціальні функції для спільного досвіду, розширену аналітику для оптимізації виставок, гнучкі варіанти монетизації, нижчі бар'єри входу через браузерний підхід. Порівняльне тестування показало 40% кращу продуктивність рендерингу порівняно з найближчими конкурентами та 60% меншу затримку для багатокористувацьких взаємодій.

Масштабованість платформи тестувалася через моделювання великих виставкових подій з тисячами одночасних користувачів. Архітектура на основі мікросервісів та контейнеризації забезпечує горизонтальне масштабування для обробки пікових навантажень. Хмарна інфраструктура AWS та Azure надає можливості автоматичного масштабування з оптимізацією витрат. Мережа доставки контенту оптимізована для 3D контенту забезпечує глобальну доступність з низькою затримкою. Стратегії розбиття бази даних та кешування підтримують операції з високою пропускною здатністю з виставковими даними та користувацькими взаємодіями.

ВИСНОВКИ

В ході виконання дипломної роботи було проведено комплексне дослідження теоретичних основ та практичних аспектів розробки платформи для віртуальних виставок у мета-всесвіті. Дослідження охопило широкий спектр питань від концептуального розуміння мета-всесвіту до технічної реалізації веб-інтерфейсу та системи навігації на основі сучасних веб-технологій.

Розроблена архітектура платформи базується на мікросервісному підході, що забезпечує масштабованість, надійність та можливість поетапного розвитку функціональності. Ключовими компонентами системи є модуль 3D-рендерингу на основі WebGL та Three.js, система управління мультимедійним контентом, модуль соціальної взаємодії та комплексна система аналітики. Така архітектура дозволяє ефективно обробляти складні 3D-сцени, забезпечувати взаємодію множинних користувачів та підтримувати різноманітні типи експонатів. Створене середовище для 3D-візуалізації демонструє високу ефективність та якість відображення віртуальних об'єктів. Використання сучасних технологій фізично-коректного рендерингу забезпечує реалістичне відтворення матеріалів та освітлення, що критично важливо для адекватного сприйняття творів мистецтва. Адаптивна система рівнів деталізації гарантує стабільну продуктивність на різних типах пристроїв, від потужних настільних комп'ютерів до мобільних телефонів.

Розроблений користувацький інтерфейс відповідає принципам сучасного UX/UI дизайну та забезпечує інтуїтивну взаємодію з віртуальним простором. Модульна архітектура інтерфейсу дозволяє адаптуватися до різних контекстів використання та потреб користувачів. Особлива увага приділена питанням доступності, що робить платформу придатною для використання людьми з особливими потребами. Система віртуальної навігації реалізує кілька режимів переміщення у просторі, адаптованих до різних типів пристроїв та рівнів досвіду користувачів. Орбітальна навігація забезпечує детальний огляд експонатів,

Комплексне тестування підтвердило готовність платформи до практичного впровадження. Модульне та інтеграційне тестування виявило високу надійність програмного коду, навантажувальне тестування продемонструвало здатність системи обробляти значну кількість одночасних користувачів. Тестування зручності використання з реальними користувачами показало високий рівень задоволеності та ефективності взаємодії з платформою. Економічний аналіз впровадження платформи показав її комерційну життєздатність та потенціал для створення нових бізнес-моделей у культурній сфері. Для музеїв та галерей платформа дозволяє значно розширити аудиторію та зменшити операційні витрати, для освітніх закладів – створити нові форми навчального досвіду. Розраховані показники повернення інвестицій демонструють привабливість проекту для потенційних інвесторів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мартим'янов А.С. Біржовий ринок криптоактивів як феномен сучасної інформаційної економіки: бакалаврська робота. – Суми, 2022. – URL: https://essuir.sumdu.edu.ua/handle/123456789/88411 (дата звернення: 12.05.2025).
2. Іванов Г.І. Дослідження віртуальних галерей та аукціонів XXI століття: дипломна робота. – Київ, 2021. – URL: https://elib.nakkkim.edu.ua/handle/123456789/4596 (дата звернення: 12.05.2025).
3. Сапогова О. Сценарно-стратегічний аналіз та планування в системі управління розвитком бренду. – Київ, 2021. – URL: https://ur.knute.edu.ua/items/c75e9cbb-e1c7-4909-8379-90807e6b8d4b (дата звернення: 12.05.2025).
4. Цуп А. Проектування продукту цифрового туризму міста Києва: дипломна робота. – Київ, 2023. – URL: https://dspace.krok.edu.ua/bitstreams/03b34b66-fb0f-4865-8eeb-99aa84c6c578/download (дата звернення: 12.05.2025).
5. Ціомік І. Платформа для організації та проведення онлайн тематичних вечірок: дипломна робота. – Київ, 2022. – URL: http://dp.knute.edu.ua/xmlui/handle/123456789/9029?show=full (дата звернення: 12.05.2025).
6. Настенко О.П. Розробка системи підтримки прийняття рішень для інтернет-платформ організації масових заходів: кваліфікаційна робота магістра. – Харків, 2023. – URL: https://openarchive.nure.ua/entities/publication/1d535caa-fd26-4fbd-aa20-32ac491770ae/full (дата звернення: 12.05.2025).
7. Меший Т.С. Візуалізація 3D моделі майданчика для проведення віртуальних конференцій: бакалаврська робота. – Суми, 2021. – URL: https://essuir.sumdu.edu.ua/bitstream/123456789/84198/1/bachelor_thesis_Meshy.pdf (дата звернення: 12.05.2025).
8. Трунов О.М., Скороїд М.Ю., Мартиненко В.С. Розвиток архітектури модулів автоматизованих систем керування приладів реабілітації. – Ужгород, 2022. – URL: https://economyandsociety.in.ua/index.php/journal/article/view/2324 (дата звернення: 12.05.2025).
9. Петросян А.Р., Петросян Р.В., Колос К.Р. Розробка платформи віддаленого управління інфраструктурою Інтернет речей. – Київ, 2021. – URL: https://bank.gov.ua/admin_uploads/article/Rebryk_2021-29-05.pdf?v=12 (дата звернення: 12.05.2025).

10. Оксамитна Л., Пряха Р. Особливості сучасних ERP-систем управління бізнес-процесами підприємства. – Київ, 2021. – URL: https://ur.knute.edu.ua/items/c75e9cbb-e1c7-4909-8379-90807e6b8d4b (дата звернення: 12.05.2025).
11. Сусленко М.О. Програмна платформа для закладів харчування на основі 3D візуалізації: дипломна робота. – Одеса, 2023. – URL: https://ela.kpi.ua/items/886a86d4-2193-439a-a4fb-14dbdbfa17e3 (дата звернення: 12.05.2025).
12. Мікульський В.В. Програмний засіб візуалізації руху об'єктів при розробці 3D симуляторів: магістерська дисертація. – Київ, 2023. – URL: https://er.nau.edu.ua/handle/NAU/64958 (дата звернення: 12.05.2025).
13. Мосиса А.А. Дослідження особливостей роботи з 3D об'єктами на мобільних платформах: магістерська робота. – Київ, 2022. – URL: https://ela.kpi.ua/bitstream/123456789/63697/1/Mosysa_magistr.pdf (дата звернення: 12.05.2025).
14. Яковенко М.О. Переваги використання CMS-систем у процесі інтеграції мультимедійного контенту на сайті (на прикладі ПП «ПОЛТАВА ТВ»): дипломна робота. – Полтава, 2025. – URL: https://reposit.nupp.edu.ua/bitstream/PoltNTU/18245/1/%d0%94%d0%b8%d0%bf%d0%bb%d0%be%d0%bc_%d0%94%d1%80%d1%83%d0%ba.%d0%af%d0%ba%d0%be%d0%b2%d0%b5%d0%bd%d0%ba%d0%be.pdf (дата звернення: 12.05.2025).
15. Корнійчук А.В. Використання штучного інтелекту для покращення SEO просування та контент маркетингу: магістерська дисертація. – Тернопіль, 2024. – URL: https://elartu.tntu.edu.ua/bitstream/lib/46958/1/Кваліфікаційна_робота_Корнійчук.pdf (дата звернення: 12.05.2025).
16. Клименко Г.П. Інтелектуальна система оцінювання якості мультимедійного контенту: кваліфікаційна робота магістра. – Миколаїв, 2024. – URL: https://krs.chmnu.edu.ua/jspui/handle/123456789/3296 (дата звернення: 12.05.2025).
17. Подран К.О. Управління проектами у сфері освіти: створення та організація електронної навчальної платформи: магістерська робота. – Луцьк, 2024. – URL: https://evnuir.vnu.edu.ua/handle/123456789/26760 (дата звернення: 12.05.2025).
18. Ягольник Д.Г. Методика покращення користувацьких характеристик веб-інтерфейсів // Актуальні задачі сучасних технологій: тези доп. 6-ї Міжнар. наук.-техн. конф. молодих учених та студентів. – Тернопіль, 2017. – Вип. 2. – С. 195. – URL: https://elartu.tntu.edu.ua/handle/lib/23010 (дата звернення: 12.05.2025).

19. Атакулова К.Ю. Дослідження методів проектування інтерфейсу веб-сайтів для забезпечення доступності користувачам з обмеженими можливостями: кваліфікаційна робота бакалавра. – Харків, 2023. – URL: https://openarchive.nure.ua/bitstreams/d3e82331-68f9-49d4-9da4-29674aa29264/download (дата звернення: 12.05.2025).
20. Козел О.Д. Дослідження методів оцінок ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережевих технологій: кваліфікаційна робота магістра. – Харків, 2024. – URL: https://openarchive.nure.ua/server/api/core/bitstreams/cfa6f96c-62f7-4f7a-a594-db35ba7cb75c/content (дата звернення: 12.05.2025).
21. Слободенюк І.Ф. Дослідження впливу візуального контенту на ефективність та користувацький досвід у веб-дизайні: кваліфікаційна робота бакалавра. – Харків, 2025. – URL: https://openarchive.nure.ua/bitstreams/e2e39a59-3a81-4908-a190-021eefa783f7/download (дата звернення: 12.05.2025).
22. Полякова Ю.А. Розробка інформаційного ресурсу для віртуального ознайомлення з університетом: кваліфікаційна робота бакалавра. – Київ, 2024. – URL: https://lib.iitta.gov.ua/id/eprint/1115/1/Grybyuk-tezy.pdf (дата звернення: 12.05.2025).
23. Радченко Д.О. Симулятор криптовалютних інвестицій. Розробка iOS-додатку для віртуальної торгівлі та прогнозування цін: кваліфікаційна робота магістра. – Київ, 2024. – URL: https://ela.kpi.ua/items/c5a580b1-5331-4ab0-a75e-c1d02ff4e8cf (дата звернення: 12.05.2025).
24. Воропаєв С.О. Навігація з використанням технологій доповненої реальності та GPS: кваліфікаційна робота бакалавра. – Запоріжжя, 2021. – URL: https://dspace.znu.edu.ua/xmlui/handle/12345/5651 (дата звернення: 12.05.2025).
25. Дмитрієв В.С. Розробка сайту з продажу нерухомості засобами React та Node.js: кваліфікаційна робота бакалавра. – Запоріжжя, 2022. – URL: https://dspace.znu.edu.ua/xmlui/handle/12345/10715 (дата звернення: 12.05.2025).
26. Суркова О. Вплив цифрових інновацій на управління персоналом IT-компанії: кваліфікаційна робота магістра. – Кропивницький, 2024. – URL: https://dspace.kntu.kr.ua/items/dbe99132-29db-42b6-b568-ebb7e4875fed (дата звернення: 12.05.2025).
27. Скорик І.І. Управління навантаженням промислового підприємства по прогнозному значенню: кваліфікаційна робота магістра. – Київ, 2024. – URL: https://ela.kpi.ua/items/0cc94e59-c692-4289-b0b9-8459e7db3247 (дата звернення: 12.05.2025).
28. Звягінцев К.А. Оцінка ефективності впровадження штучного інтелекту в бізнес-процеси підприємства: кваліфікаційна робота магістра. – Київ,

2024. – URL: https://ela.kpi.ua/items/6a0fc446-a52e-48e9-a5d6-bca70a545f95 (дата звернення: 12.05.2025).
29. Комарницький М.В. Розроблення веб-застосунку для забезпечення вибору дисциплін здобувачами КПНУ ім. Огієнка: кваліфікаційна робота бакалавра. – Кам'янець-Подільський, 2024. – URL: https://dspace.nuft.edu.ua/server/api/core/bitstreams/1bb9b9d8-8b39-4ce1-9874-5fc19859e7af/content (дата звернення: 12.05.2025).
30. Чичикало Є.А. Віртуальний тренажер "Установка кутів прицілювання прицілу ПГ-4. Вертикальна наводка": кваліфікаційна робота бакалавра. – Суми, 2021. – URL: https://essuir.sumdu.edu.ua/bitstream/123456789/84173/1/Chychykalo_bac_rob.pdf (дата звернення: 12.05.2025).
31. Костенко Д. Впровадження віртуального інформаційного середовища у освітній процес: магістерська робота. – Київ, 2023. – URL: https://www.academia.edu/105529136 (дата звернення: 12.05.2025).
32. Гончарова Н. Технологія доповненої реальності в підручниках нового покоління: дис. … канд. пед. наук. – Київ, 2024. – URL: https://lib.iitta.gov.ua/id/eprint/716685/1/9c8b6a35b1ea5b7130c1ae9942824e97.pdf (дата звернення: 12.05.2025).
33. Котельникова Ю.М. Підвищення конкурентоспроможності підприємств в умовах цифровізації: кваліфікаційна робота магістра. – Харків, 2022. – URL: https://scispace.com/pdf/increasing-the-competitiveness-of-enterprises-under-the-30a2gdku.pdf (дата звернення: 12.05.2025).
34. Чміль А.В. Автоматизація HR-функцій у малому бізнесі (на прикладі ФОП Толстигін І.Л., м. Хмельницький): кваліфікаційна робота магістра. – Хмельницький, 2024. – URL: https://elar.khmnu.edu.ua/bitstreams/71886c3a-7eb9-446d-b47a-09c63e3ef81a/download (дата звернення: 12.05.2025).
35. Буртник І.Л. Розробка банківського сервісу для перегляду стану рахунків та переказу коштів із застосуванням бібліотеки ReactJS: кваліфікаційна робота бакалавра. – Тернопіль, 2024. – URL: https://elartu.tntu.edu.ua/handle/lib/45447 (дата звернення: 12.05.2025).
36. Воржакова Ю.П., Лінива І.С. Формування моделі мотивації праці для зростання професійного інтелекту команди: кваліфікаційна робота магістра. – Київ, 2022. – URL: https://ela.kpi.ua/bitstreams/2d55a07f-b687-49af-857f-4f24ba501e4a/download (дата звернення: 12.05.2025).
37. Крівель М. Діджиталізація діяльності підприємства в умовах воєнного стану: кваліфікаційна робота магістра. – Київ, 2023. – URL: https://ela.kpi.ua/items/a375784e-2deb-457a-9e8a-78913bbbe6ab (дата звернення: 12.05.2025).

38. Домашенко С. Діджиталізація бізнесу в умовах воєнного стану: кваліфікаційна робота магістра. – Київ, 2024. – URL: https://www.academia.edu/112625359 (дата звернення: 12.05.2025).
39. Пригодій М. Інформаційні технології в нових умовах освітнього процесу і управління ним: магістерська робота. – Львів, 2024. – URL: https://www.researchgate.net/publication/373311522 (дата звернення: 12.05.2025).
40. Форсюк А. Кваліфікаційна робота з розробки телеграм бота для комерції: магістерська робота. – Київ, 2024. – URL: https://dspace.nuft.edu.ua/server/api/core/bitstreams/08593c5d-8ab9-4bbd-b047-da0e7b6d26f0/content (дата звернення: 12.05.2025).

ДОДАТКИ

Додаток А

Проект працює по ссилці https://meta1-71cm.onrender.com
Логін:admin
Пароль:admin
рух на w/a/s/d (якщо не рухаєтесь то переключіть на англ мову або рухайтесь стрілочками)
клавіша esc показує курсор далі наводитесь на обєкт і він вам показує опис.

Додаток Б

Admin.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Admin Panel</title>
  <style>
    body {
      font-family: 'Segoe UI', Arial, sans-serif;
      background: linear-gradient(120deg, #e0e7ff 0%, #f8fafc 100%);
      min-height: 100vh;
      margin: 0;
    }
    .panel-container {
      max-width: 900px;
      margin: 4rem auto;
      background: #f7faff;
      border-radius: 24px;
      box-shadow: 0 8px 32px rgba(60,80,180,0.14);
```

```
  padding: 2.5rem 2.5rem 2.2rem 2.5rem;
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 2.2rem;
}
form#loginForm, #objectPanel {
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 1.5rem;
}
.fade-in {
  opacity: 1;
  transform: translateY(0);
  transition: opacity 0.5s, transform 0.5s;
}
.fade-out {
  opacity: 0;
  transform: translateY(-40px);
  pointer-events: none;
  transition: opacity 0.5s, transform 0.5s;
}
h1 {
  text-align: center;
  color: #1a237e;
  margin-bottom: 1.2rem;
```

```
  font-size: 2.1rem;
  letter-spacing: 1px;
}
.object-list {
  margin-top: 0.5rem;
  display: flex;
  flex-direction: column;
  gap: 1.2rem;
  width: 100%;
  align-items: center;
}
.object-item {
  background: linear-gradient(90deg, #e0e7ff 60%, #f1f5f9 100%);
  border-radius: 14px;
  padding: 1.1rem 1.5rem;
  width: 760px;
  min-width: 760px;
  max-width: 760px;
  box-shadow: 0 2px 8px rgba(60,80,180,0.08);
  display: flex;
  flex-direction: column;
  gap: 0.4rem;
  cursor: pointer;
  border: 2.5px solid transparent;
  transition: border 0.2s, background 0.2s;
  align-items: center;
  font-size: 1.09rem;
  text-align: center;
```

```
  box-sizing: border-box;
  word-break: break-word;
  overflow-wrap: break-word;
  white-space: normal;
  text-overflow: ellipsis;
  max-height: 180px;
  overflow-y: auto;
}
.object-item.selected {
  border: 2.5px solid #6366f1;
  background: linear-gradient(90deg, #dbeafe 60%, #e0e7ff 100%);
  box-shadow: 0 4px 16px rgba(60,80,180,0.13);
}
.object-item code {
  color: #6366f1;
  background: #e0e7ff;
  border-radius: 6px;
  padding: 0.13rem 0.6rem;
  font-size: 1em;
  margin-bottom: 0.2rem;
  display: inline-block;
}
.object-item .type {
  color: #1e40af;
  font-size: 0.98em;
  font-weight: 600;
  margin-bottom: 0.2rem;
  letter-spacing: 0.2px;
```

```
}
.object-item span {
  display: block;
  width: 100%;
  word-break: break-word;
  overflow-wrap: break-word;
  white-space: normal;
  text-align: center;
}
.save-btn {
  background: linear-gradient(90deg, #6366f1 60%, #1e40af 100%);
  color: #fff;
  border: none;
  border-radius: 12px;
  padding: 0.9rem 1.7rem;
  font-size: 1.15rem;
  font-weight: 600;
  cursor: pointer;
  margin-top: 1rem;
  box-shadow: 0 2px 8px rgba(60,80,180,0.11);
  transition: background 0.2s, box-shadow 0.2s;
  width: 100%;
  letter-spacing: 0.5px;
}
.save-btn:hover {
  background: linear-gradient(90deg, #4f46e5 60%, #1e3a8a 100%);
  box-shadow: 0 4px 16px rgba(60,80,180,0.17);
}
```

```
.success-msg {
  color: #16a34a;
  text-align: center;
  margin-top: 1rem;
  font-weight: 500;
}
input[type="text"], input[type="password"] {
  width: 90%;
  margin-left: auto;
  margin-right: auto;
  padding: 0.9rem 1.2rem;
  border-radius: 12px;
  border: 1.7px solid #c7d2fe;
  font-size: 1.15rem;
  margin-bottom: 0.8rem;
  outline: none;
  transition: border 0.2s;
  background: #f8fafc;
  text-align: center;
}
input[type="text"]:focus, input[type="password"]:focus {
  border: 1.7px solid #6366f1;
  background: #e0e7ff;
}
/* Плавний виїзд форми логіну */
#loginForm {
  transform: translateY(60px);
  opacity: 0;
```

```
    animation: slideInLogin 0.7s cubic-bezier(.4,1.4,.6,1) 0.1s forwards;
   }
   @keyframes slideInLogin {
    from {
     transform: translateY(60px);
     opacity: 0;
    }
    to {
     transform: translateY(0);
     opacity: 1;
    }
   }
   /* Центрування блоку логіну */
   #loginForm > div {
    width: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
   }
  </style>
</head>
<body>
  <div class="panel-container">
   <form id="loginForm" class="fade-in" autocomplete="off" onsubmit="return
handleLogin(event)">
    <h1>Вхід адміністратора</h1>
    <div style="margin-bottom:1.2rem;">
     <input type="text" id="login" placeholder="Логін" required autofocus>
```

```
      <input type="password" id="password" placeholder="Пароль" required>
    </div>
    <button type="submit" class="save-btn" style="margin-top:0;">Увійти</button>
    <div class="success-msg" id="loginMsg" style="color:#dc2626;"></div>
  </form>
  <div id="objectPanel" class="fade-out" style="display:none;">
    <h1>Вибір 3D-об'єкта</h1>
    <div class="object-list" id="object-list"></div>
    <button class="save-btn" onclick="saveSelection()">Зберегти вибір</button>
    <div class="success-msg" id="successMsg"></div>
  </div>
</div>
<script>
  // --- Динамічне завантаження моделей з папки ---
  const modelFiles = [
    'models/Лисиця.glb',
    'models/Ноутбук.glb',
    'models/Відеокарта.glb'
  ];
  const objects = [
    {
      name: 'Лисиця',
      description: 'Мудра лисиця — символ гнучкості, винахідливості та
цифрової свободи у віртуальному просторі.',
      type: 'Тварина (Animal)',
      url: 'models/Лисиця.glb'
    },
    {
      name: 'Ноутбук',
```

```
    description: 'Сучасний ноутбук — інструмент для навчання, роботи та творчості у цифровому світі.',
    type: 'Техніка (Laptop)',
    url: 'models/Ноутбук.glb'
  },
  {
    name: 'Відеокарта',
    description: 'Потужна відеокарта — серце графічних обчислень і 3D-візуалізації у метавсесвіті.',
    type: 'Техніка (GPU)',
    url: 'models/Відеокарта.glb'
  }
];
let selectedIdxs = [];
function renderList() {
  const list = document.getElementById('object-list');
  list.innerHTML = objects.map((o, i) => `
    <div class="object-item${selectedIdxs.includes(i) ? ' selected' : ''}" onclick="toggleObject(${i})">
      <strong>${o.name}</strong>
      <div class="type">${o.type}</div>
      <code>${o.url}</code>
      <span>${o.description}</span>
    </div>
  `).join('');
}
function toggleObject(idx) {
  if (selectedIdxs.includes(idx)) {
    selectedIdxs = selectedIdxs.filter(i => i !== idx);
```

```
  } else {
    selectedIdxs.push(idx);
  }
  renderList();
  document.getElementById('successMsg').textContent = '';
}
function saveSelection() {
  if (selectedIdxs.length === 0) {
    alert('Оберіть хоча б один об\'єкт!');
    return;
  }
  const selected = selectedIdxs.map(i => objects[i]);
  localStorage.setItem('selectedObjects', JSON.stringify(selected));
  document.getElementById('successMsg').textContent = 'Вибір збережено!';
  setTimeout(() => {
    window.location.href = 'viewer.html';
  }, 700); // Плавний перехід через 0.7 сек
}
// --- АВТОРИЗАЦІЯ ---
function handleLogin(e) {
  e.preventDefault();
  const login = document.getElementById('login').value.trim();
  const pass = document.getElementById('password').value;
  const msg = document.getElementById('loginMsg');
  if (login === 'admin' && pass === 'admin') {
    // Анімація: приховати форму, показати панель вибору
    document.getElementById('loginForm').classList.remove('fade-in');
    document.getElementById('loginForm').classList.add('fade-out');
```

```
        setTimeout(() => {
          document.getElementById('loginForm').style.display = 'none';
          const objPanel = document.getElementById('objectPanel');
          objPanel.style.display = '';
          objPanel.classList.remove('fade-out');
          objPanel.classList.add('fade-in');
          renderList();
        }, 500);
      } else {
        msg.textContent = 'Невірний логін або пароль';
      }
      return false;
    }
    // --- /АВТОРИЗАЦІЯ ---
  </script>
</body>
</html>
```

Додаток В

viewer.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>3D Viewer</title>
  <style>
    body { margin: 0; overflow: hidden; }
    canvas { display: block; }
```

```
    #description {
      position: absolute;
      top: 10px;
      left: 10px;
      background: rgba(0, 0, 0, 0.7);
      color: white;
      padding: 10px;
      border-radius: 5px;
      display: none;
    }
    #crosshair {
      position: absolute;
      left: 50%;
      top: 50%;
      width: 12px;
      height: 12px;
      margin-left: -6px;
      margin-top: -6px;
      pointer-events: none;
      z-index: 10;
      border-radius: 50%;
      background: rgba(0,0,0,0.2);
      border: 2px solid #fff;
      box-sizing: border-box;
    }
  </style>
</head>
<body>
```

```
  <div id="description"></div>
  <div id="crosshair"></div>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/three.js/r128/three.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/three/examples/js/loaders/GLTFLoader.js"></script
>
  <script
src="https://cdn.jsdelivr.net/npm/three/examples/jsm/libs/fontloader.js"></script>
  <script>
    // Створення сцени
    const scene = new THREE.Scene();
    scene.background = new THREE.Color(0xeeeeee);

    // Розміри кімнати (збільшено)
    const roomWidth = 100;
    const roomHeight = 50;
    const roomDepth = 100;

    // ПІДЛОГА: світло-бежевий колір, без текстури
    const floorMaterial = new THREE.MeshStandardMaterial({ color: 0xf7ecd0 }); //
пастельний бежевий
    const floorGeometry = new THREE.PlaneGeometry(roomWidth, roomDepth, 1,
1);
    const floor = new THREE.Mesh(floorGeometry, floorMaterial);
    floor.rotation.x = -Math.PI / 2;
    floor.position.y = -roomHeight / 2;
    scene.add(floor);

    // СТІНИ: ще більш приємний пастельний колір (наприклад, світло-
лавандовий)
```

```
    const wallColor = 0xe8eaff; // дуже ніжний лавандовий
    const wallMaterial = new THREE.MeshStandardMaterial({ color: wallColor, side: THREE.DoubleSide });
    // Задня стіна
    const backWall = new THREE.Mesh(new THREE.PlaneGeometry(roomWidth, roomHeight), wallMaterial);
    backWall.position.z = -roomDepth / 2;
    backWall.position.y = 0;
    scene.add(backWall);
    // Передня стіна
    const frontWall = new THREE.Mesh(new THREE.PlaneGeometry(roomWidth, roomHeight), wallMaterial);
    frontWall.position.z = roomDepth / 2;
    frontWall.position.y = 0;
    frontWall.rotation.y = Math.PI;
    scene.add(frontWall);
    // Ліва стіна
    const leftWall = new THREE.Mesh(new THREE.PlaneGeometry(roomDepth, roomHeight), wallMaterial);
    leftWall.position.x = -roomWidth / 2;
    leftWall.position.y = 0;
    leftWall.rotation.y = Math.PI / 2;
    scene.add(leftWall);
    // Права стіна (з проходом для дверей)
    (function() {
      const doorWidth = 16;
      const wallPartWidth = (roomDepth - doorWidth) / 2;
      const wallHeight = roomHeight;
      const rightWallLeftGeo = new THREE.PlaneGeometry(wallPartWidth, wallHeight);
```

```
    const rightWallLeft = new THREE.Mesh(rightWallLeftGeo, wallMaterial);
    rightWallLeft.position.x = roomWidth/2;
    rightWallLeft.position.y = 0;
    rightWallLeft.position.z = -((doorWidth/2) + wallPartWidth/2);
    rightWallLeft.rotation.y = -Math.PI/2;
    scene.add(rightWallLeft);
    // Середня частина (прохід) - не додаємо
    // Права вертикальна частина
    const rightWallRightGeo = new THREE.PlaneGeometry(wallPartWidth,
wallHeight);
    const rightWallRight = new THREE.Mesh(rightWallRightGeo, wallMaterial);
    rightWallRight.position.x = roomWidth/2;
    rightWallRight.position.y = 0;
    rightWallRight.position.z = (doorWidth/2) + wallPartWidth/2;
    rightWallRight.rotation.y = -Math.PI/2;
    scene.add(rightWallRight);
  })();

  // --- Прохід між кімнатами: суцільна підлога між кімнатами ---
  (function() {
    const doorWidth = 16;
    const passageDepth = 8;
    const totalPassage = (roomWidth/2) + passageDepth + (roomWidth/2);
    const floorGeo = new THREE.PlaneGeometry(passageDepth, doorWidth);
    // Підлога між кімнатами
    const floorMesh = new THREE.Mesh(
      new THREE.PlaneGeometry(passageDepth * 2 + 0.1, doorWidth),
      new THREE.MeshStandardMaterial({ color: 0xf7ecd0 })
    );
```

```
  floorMesh.rotation.x = -Math.PI/2;
  floorMesh.position.set((roomWidth/2) + passageDepth/2 + 10, -roomHeight/2, 0);
  scene.add(floorMesh);
})();

// Завантаження обраних об'єктів з localStorage
let selectedObjects = [];
try {
  selectedObjects = JSON.parse(localStorage.getItem('selectedObjects')) || [];
} catch (e) {}
console.log('Loaded objects from localStorage:', selectedObjects);

// Масив для збереження mesh-об'єктів для опису
const loadedObjects = [];

// Додаємо всі вибрані об'єкти у кімнату
if (selectedObjects.length > 0) {
  selectedObjects.forEach((obj, i) => {
    const angle = (i / selectedObjects.length) * Math.PI * 2;
    const radius = 20;
    const x = Math.cos(angle) * radius;
    const z = Math.sin(angle) * radius;
    const y = -roomHeight / 2 + 3;

    // Пілар (циліндр)
    const pillarHeight = 6;
    const pillarGeometry = new THREE.CylinderGeometry(2, 2, pillarHeight, 32);
    const pillarMaterial = new THREE.MeshStandardMaterial({ color: 0xf0f0f0 });
```

```
const pillar = new THREE.Mesh(pillarGeometry, pillarMaterial);
pillar.position.set(x, y + pillarHeight / 2 + 6, z);
scene.add(pillar);

// Додаємо об'єкт на пілар трохи вище
if (obj.url && obj.url.match(/\.(gltf|glb)$/i)) {
  const loader = new THREE.GLTFLoader();
  loader.load(obj.url, (gltf) => {
    const mesh = gltf.scene;
    mesh.position.set(x, y + pillarHeight + 10, z);
    if (obj.url && obj.url.match(/Ноутбук\.glb$/i)) {
      mesh.scale.set(7, 7, 7); // ще більший масштаб для ноутбука
    } else if (obj.url && obj.url.match(/Відеокарта\.glb$/i)) {
      mesh.scale.set(24, 24, 24); // масштаб для відеокарти
    } else if (obj.url && obj.url.match(/Лисиця\.glb$/i)) {
      mesh.scale.set(0.03, 0.03, 0.03); // Стандартний масштаб для лисиці
    } else {
      mesh.scale.set(0.03, 0.03, 0.03);
    }

    // Додаємо опис та значно збільшуємо область взаємодії
    mesh.traverse((child) => {
      if (child.isMesh) {
        child.userData = {
          description: obj.description || 'Об\'єкт без опису',
          type: obj.type || 'Невідомий тип',
        };
        child.geometry.computeBoundingBox();
```

```
            const boundingBox = child.geometry.boundingBox;
            const interactionScale = 30; // Дуже велика область взаємодії
            boundingBox.expandByScalar(interactionScale);
          }
        });

        scene.add(mesh);
        loadedObjects.push(mesh);
      }, undefined, (err) => {
        console.error(`Помилка завантаження моделі ${obj.name}:`, err);
      });
    } else {
      console.warn(`Пропущено об\'єкт ${obj.name} через невірний URL:`, obj.url);
    }
  });
} else {
  console.warn('Жоден об\'єкт не було обрано для завантаження.');
}

// Додаємо вікна у стінах
function addWindow(x, y, z, rotY = 0, w = 16, h = 10) {
  const windowGeo = new THREE.PlaneGeometry(w, h);
  const windowMat = new THREE.MeshBasicMaterial({ color: 0xbfdfff, transparent: true, opacity: 0.35 });
  const win = new THREE.Mesh(windowGeo, windowMat);
  win.position.set(x, y, z);
  win.rotation.y = rotY;
  scene.add(win);
```

```
    }
    // Вікно у задній стіні
    addWindow(0, 0, -roomDepth/2 + 0.1);
    // Вікно у передній стіні
    addWindow(0, 0, roomDepth/2 - 0.1, Math.PI);
    // Вікно у лівій стіні
    addWindow(-roomWidth/2 + 0.1, 0, 0, Math.PI/2);
    // Вікно у правій стіні (видалено, бо там табличка)
    // addWindow(roomWidth/2 - 0.1, 0, 0, -Math.PI/2);

    // --- Переносимо табличку з інформацією на стіну другої кімнати ---
    function addInfoBoard() {
      const boardW = 32, boardH = 14;
      const boardGeo = new THREE.PlaneGeometry(boardW, boardH);
      const boardMat = new THREE.MeshBasicMaterial({ color: 0xffffff, transparent:
true, opacity: 0.92 });
      const offsetX = 120;
      const board = new THREE.Mesh(boardGeo, boardMat);
      board.position.set(offsetX + roomWidth/2 - 0.2, 8, 0); // на правій стіні другої
кімнати
      board.rotation.y = -Math.PI/2;
      scene.add(board);
      // Додаємо текст (canvas texture)
      const canvas = document.createElement('canvas');
      canvas.width = 1024; canvas.height = 448;
      const ctx = canvas.getContext('2d');
      ctx.fillStyle = '#fff'; ctx.fillRect(0,0,canvas.width,canvas.height);
      ctx.fillStyle = '#222';
      ctx.font = 'bold 54px Segoe UI, Arial';
```

```
  ctx.fillText('Проєкт "Metaverse Room"', 40, 90);
  ctx.font = '36px Segoe UI, Arial';
  ctx.fillText('Автор: Іван Біленко', 40, 170);
  ctx.fillText('Чи це метапростір: Так', 40, 230);
  ctx.fillText('Бібліотеки: three.js, express, sqlite3', 40, 290);
  ctx.fillText('Мова: JavaScript (Node.js, frontend)', 40, 350);
  ctx.font = 'italic 32px Segoe UI, Arial';
  ctx.fillText('2025', 40, 410);
  const tex = new THREE.CanvasTexture(canvas);
  const textMat = new THREE.MeshBasicMaterial({ map: tex, transparent: true });
  const textMesh = new THREE.Mesh(boardGeo, textMat);
  textMesh.position.copy(board.position);
  textMesh.rotation.copy(board.rotation);
  scene.add(textMesh);
}
addInfoBoard();

// --- Додаємо фото ДПУ з локального файлу на лівій стіні другої кімнати ---
(function() {
  const offsetX = 120;
  const frameW = 28, frameH = 16;
  // Фото (текстура локально)
  const photoGeo = new THREE.PlaneGeometry(frameW, frameH);
  const loader = new THREE.TextureLoader();
  loader.load(
    'models/183uacover06.jpg',
    function(tex) {
      console.log('Фото ДПУ завантажено локально');
```

```
      const photoMat = new THREE.MeshBasicMaterial({ map: tex, transparent:
true });
      const photo = new THREE.Mesh(photoGeo, photoMat);
      photo.position.set(offsetX, 8, -roomDepth/2 + 1.5);
      photo.rotation.y = 0;
      photo.position.z += 0.1;
      scene.add(photo);
     },
     undefined,
     function(err) {
      console.error('Помилка завантаження фото ДПУ локально:', err);
     }
    );
    // Кнопка-посилання на стіні (3D кнопка)
    const buttonW = 12, buttonH = 3;
    const buttonGeo = new THREE.PlaneGeometry(buttonW, buttonH);
    const buttonCanvas = document.createElement('canvas');
    buttonCanvas.width = 512; buttonCanvas.height = 128;
    const bctx = buttonCanvas.getContext('2d');
    bctx.fillStyle = '#fff';
    bctx.fillRect(0,0,512,128);
    bctx.fillStyle = '#1a237e';
    bctx.font = 'bold 48px Segoe UI, Arial';
    bctx.textAlign = 'center';
    bctx.textBaseline = 'middle';
    bctx.fillText('Перейти на сайт ДПУ', 256, 64);
    const buttonTex = new THREE.CanvasTexture(buttonCanvas);
    const buttonMat = new THREE.MeshBasicMaterial({ map: buttonTex });
    const button = new THREE.Mesh(buttonGeo, buttonMat);
```

```
    button.position.set(offsetX, -2, -roomDepth/2 + 1.5); // нижче фото (було 2, стало -2)
    button.rotation.y = 0;
    button.cursor = 'pointer';
    scene.add(button);
    // Взаємодія: клік по кнопці відкриває сайт
    const raycaster = new THREE.Raycaster();
    const mouse = new THREE.Vector2();
    function onClick(event) {
      mouse.x = (event.clientX / window.innerWidth) * 2 - 1;
      mouse.y = -(event.clientY / window.innerHeight) * 2 + 1;
      raycaster.setFromCamera(mouse, camera);
      const intersects = raycaster.intersectObject(button);
      if (intersects.length > 0) {
        window.open('https://dpu.edu.ua/', '_blank');
      }
    }
    window.addEventListener('click', onClick);
  })();

  // --- Додаємо другу кімнату поряд з основною ---
  (function addSecondRoom() {
    const offsetX = 120; // Відстань між кімнатами
    // Підлога
    const floor2 = new THREE.Mesh(
      new THREE.PlaneGeometry(roomWidth, roomDepth, 1, 1),
      new THREE.MeshStandardMaterial({ color: 0xe0e0e0 })
    );
    floor2.rotation.x = -Math.PI / 2;
```

```
    floor2.position.set(offsetX, -roomHeight / 2, 0);
    scene.add(floor2);
    // Стіни
    const wallMat2 = new THREE.MeshStandardMaterial({ color: 0xd0eaff, side: THREE.DoubleSide });
    const backWall2 = new THREE.Mesh(new THREE.PlaneGeometry(roomWidth, roomHeight), wallMat2);
    backWall2.position.set(offsetX, 0, -roomDepth / 2);
    scene.add(backWall2);
    const frontWall2 = new THREE.Mesh(new THREE.PlaneGeometry(roomWidth, roomHeight), wallMat2);
    frontWall2.position.set(offsetX, 0, roomDepth / 2);
    frontWall2.rotation.y = Math.PI;
    scene.add(frontWall2);
  })();

  // --- Друга кімната: задня стіна з проходом (3 вертикальні частини) ---
  (function() {
    const offsetX = 120;
    const doorWidth = 16;
    const wallPartWidth = (roomWidth - doorWidth) / 2;
    const wallHeight = roomHeight;
    const wallMat2 = new THREE.MeshStandardMaterial({ color: 0xd0eaff, side: THREE.DoubleSide });
    // Ліва вертикальна частина
    const backWall2LeftGeo = new THREE.PlaneGeometry(wallPartWidth, wallHeight);
    const backWall2Left = new THREE.Mesh(backWall2LeftGeo, wallMat2);
    backWall2Left.position.set(offsetX - (roomWidth/2) + wallPartWidth/2, 0, -roomDepth / 2);
```

```
    scene.add(backWall2Left);
    // Середня частина (прохід) - не додаємо
    // Права вертикальна частина
    const backWall2RightGeo = new THREE.PlaneGeometry(wallPartWidth, wallHeight);
    const backWall2Right = new THREE.Mesh(backWall2RightGeo, wallMat2);
    backWall2Right.position.set(offsetX + (roomWidth/2) - wallPartWidth/2, 0, -roomDepth / 2);
    scene.add(backWall2Right);
  })();

  // --- Друга кімната: права стіна (суцільна, без проходу) ---
  (function() {
    const offsetX = 120;
    const wallMat2 = new THREE.MeshStandardMaterial({ color: 0xd0eaff, side: THREE.DoubleSide });
    const rightWall2 = new THREE.Mesh(new THREE.PlaneGeometry(roomDepth, roomHeight), wallMat2);
    rightWall2.position.x = offsetX + roomWidth/2;
    rightWall2.position.y = 0;
    rightWall2.position.z = 0;
    rightWall2.rotation.y = -Math.PI/2;
    scene.add(rightWall2);
  })();

  // --- Друга кімната: ліва стіна з проходом (3 вертикальні частини) ---
  (function() {
    const offsetX = 120;
    const doorWidth = 16;
    const wallPartWidth = (roomDepth - doorWidth) / 2;
```

```
    const wallHeight = roomHeight;
    const wallMat2 = new THREE.MeshStandardMaterial({ color: 0xd0eaff, side: THREE.DoubleSide });
    // Ліва вертикальна частина
    const leftWall2LeftGeo = new THREE.PlaneGeometry(wallPartWidth, wallHeight);
    const leftWall2Left = new THREE.Mesh(leftWall2LeftGeo, wallMat2);
    leftWall2Left.position.x = offsetX - roomWidth/2;
    leftWall2Left.position.y = 0;
    leftWall2Left.position.z = -((doorWidth/2) + wallPartWidth/2);
    leftWall2Left.rotation.y = Math.PI/2;
    scene.add(leftWall2Left);
    // Середня частина (прохід) - не додаємо
    // Права вертикальна частина
    const leftWall2RightGeo = new THREE.PlaneGeometry(wallPartWidth, wallHeight);
    const leftWall2Right = new THREE.Mesh(leftWall2RightGeo, wallMat2);
    leftWall2Right.position.x = offsetX - roomWidth/2;
    leftWall2Right.position.y = 0;
    leftWall2Right.position.z = (doorWidth/2) + wallPartWidth/2;
    leftWall2Right.rotation.y = Math.PI/2;
    scene.add(leftWall2Right);
  })();

  // --- Додаємо освітлення у другій кімнаті ---
  (function addSecondRoomLight() {
    const offsetX = 120;
    const light2 = new THREE.PointLight(0xfff7cc, 1.2, 200);
    light2.position.set(offsetX, 30, 0);
```

```
  scene.add(light2);
})();

// Камера
const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
camera.position.set(0, 0, 8);
let cameraDirection = 0; // radians
let pitch = 0; // вертикальний кут

// --- Освітлення ---
// Додаємо просте світло
const light = new THREE.PointLight(0xffffff, 1.1, 200);
light.position.set(0, 30, 0);
scene.add(light);

// Рендерер
const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

// Опис
const raycaster = new THREE.Raycaster();
const mouse = new THREE.Vector2();
const descriptionBox = document.getElementById('description');

function onMouseMove(event) {
  mouse.x = (event.clientX / window.innerWidth) * 2 - 1;
  mouse.y = -(event.clientY / window.innerHeight) * 2 + 1;
```

```
    raycaster.setFromCamera(mouse, camera);
    const objectsToCheck = [...loadedObjects].filter(Boolean);
    const intersects = raycaster.intersectObjects(objectsToCheck, true);
    if (intersects.length > 0) {
      descriptionBox.style.display = 'block';
      descriptionBox.textContent = intersects[0].object.userData.description;
    } else {
      descriptionBox.style.display = 'none';
    }
  }
  window.addEventListener('mousemove', onMouseMove);

  // Управління камерою (WASD/стрілки)
  const keys = {};
  window.addEventListener('keydown', (e) => { keys[e.key.toLowerCase()] = true;
});
  window.addEventListener('keyup', (e) => { keys[e.key.toLowerCase()] = false; });

  // --- Pointer Lock Controls для огляду мишкою ---
  let isPointerLocked = false;
  document.body.addEventListener('click', () => {
    renderer.domElement.requestPointerLock();
  });
  document.addEventListener('pointerlockchange', () => {
    isPointerLocked = document.pointerLockElement === renderer.domElement;
  });
  document.addEventListener('mousemove', (event) => {
    if (isPointerLocked) {
      cameraDirection -= event.movementX * 0.002;
```

```
    pitch += event.movementY * 0.002; // БЕЗ інверсії
    // Обмеження вертикального кута
    pitch = Math.max(-Math.PI/2 + 0.1, Math.min(Math.PI/2 - 0.1, pitch));
  }
});

function updateCamera() {
  const speed = 0.4; // Збільшена швидкість переміщення
  // Вектор напрямку (y не змінюється)
  const dx = Math.sin(cameraDirection);
  const dz = Math.cos(cameraDirection);
  // Вперед/назад (W/S або стрілки)
  if (keys['w'] || keys['arrowup']) {
    camera.position.x -= dx * speed;
    camera.position.z -= dz * speed;
  }
  if (keys['s'] || keys['arrowdown']) {
    camera.position.x += dx * speed;
    camera.position.z += dz * speed;
  }
  // Вліво/вправо (A/D або стрілки)
  if (keys['a'] || keys['arrowleft']) {
    camera.position.x -= Math.cos(cameraDirection) * speed;
    camera.position.z += Math.sin(cameraDirection) * speed;
  }
  if (keys['d'] || keys['arrowright']) {
    camera.position.x += Math.cos(cameraDirection) * speed;
    camera.position.z -= Math.sin(cameraDirection) * speed;
```

```
    }
    // Огляд мишкою (y не змінюється)
    const lookAtX = camera.position.x - Math.sin(cameraDirection) *
Math.cos(pitch);
    const lookAtY = camera.position.y - Math.sin(pitch);
    const lookAtZ = camera.position.z - Math.cos(cameraDirection) *
Math.cos(pitch);
    camera.lookAt(lookAtX, lookAtY, lookAtZ);
  }

  // Анімація
  function animate() {
    requestAnimationFrame(animate);
    loadedObjects.forEach(obj => { if (obj && obj.rotation) obj.rotation.y += 0.01;
});
    updateCamera();
    renderer.render(scene, camera);
  }
  animate();

  // --- Додаємо фото Oracle і Cisco на стіні навпроти (передній стіні другої
кімнати) ---
  (function() {
    const offsetX = 120;
    const frameW = 12, frameH = 12; // квадратні рамки для логотипів
    const spacing = 2; // відступ між фото
    const baseY = 8; // по центру стіни
    const baseZ = roomDepth/2 - 1.5 - 0.2; // передня стіна другої кімнати
    const loader = new THREE.TextureLoader();
    // Oracle
```

```
    loader.load(
      'models/oracle.png',
      function(tex) {
        const photoMat = new THREE.MeshBasicMaterial({ map: tex, transparent:
true });
        const photoGeo = new THREE.PlaneGeometry(frameW, frameH);
        const photo = new THREE.Mesh(photoGeo, photoMat);
        photo.position.set(offsetX - frameW/2 - spacing/2, baseY, baseZ);
        photo.rotation.y = Math.PI; // Повертаємо до кімнати
        scene.add(photo);
      },
      undefined,
      function(err) { console.error('Помилка завантаження oracle.png:', err); }
    );
    // Cisco
    loader.load(
      'models/cisco.png',
      function(tex) {
        const photoMat = new THREE.MeshBasicMaterial({ map: tex, transparent:
true });
        const photoGeo = new THREE.PlaneGeometry(frameW, frameH);
        const photo = new THREE.Mesh(photoGeo, photoMat);
        photo.position.set(offsetX + frameW/2 + spacing/2, baseY, baseZ);
        photo.rotation.y = Math.PI;
        scene.add(photo);
      },
      undefined,
      function(err) { console.error('Помилка завантаження cisco.png:', err); }
    );
```

```
// Кнопки під фото
const buttonW = 10, buttonH = 2.5;
const buttonYOffset = -frameH/2 - 2;
// Oracle button
const oracleBtnCanvas = document.createElement('canvas');
oracleBtnCanvas.width = 400; oracleBtnCanvas.height = 100;
const octx = oracleBtnCanvas.getContext('2d');
octx.fillStyle = '#fff'; octx.fillRect(0,0,400,100);
octx.fillStyle = '#b71c1c';
octx.font = 'bold 36px Segoe UI, Arial';
octx.textAlign = 'center'; octx.textBaseline = 'middle';
octx.fillText('Oracle Academy', 200, 50);
const oracleBtnTex = new THREE.CanvasTexture(oracleBtnCanvas);
const oracleBtnMat = new THREE.MeshBasicMaterial({ map: oracleBtnTex });
const oracleBtnGeo = new THREE.PlaneGeometry(buttonW, buttonH);
const oracleBtn = new THREE.Mesh(oracleBtnGeo, oracleBtnMat);
oracleBtn.position.set(offsetX - frameW/2 - spacing/2, baseY + buttonYOffset,
baseZ);
oracleBtn.rotation.y = Math.PI;
oracleBtn.cursor = 'pointer';
scene.add(oracleBtn);
// Cisco button
const ciscoBtnCanvas = document.createElement('canvas');
ciscoBtnCanvas.width = 400; ciscoBtnCanvas.height = 100;
const cctx = ciscoBtnCanvas.getContext('2d');
cctx.fillStyle = '#fff'; cctx.fillRect(0,0,400,100);
cctx.fillStyle = '#1565c0';
cctx.font = 'bold 36px Segoe UI, Arial';
cctx.textAlign = 'center'; cctx.textBaseline = 'middle';
```

```
    cctx.fillText('Cisco Networking', 200, 50);
    const ciscoBtnTex = new THREE.CanvasTexture(ciscoBtnCanvas);
    const ciscoBtnMat = new THREE.MeshBasicMaterial({ map: ciscoBtnTex });
    const ciscoBtnGeo = new THREE.PlaneGeometry(buttonW, buttonH);
    const ciscoBtn = new THREE.Mesh(ciscoBtnGeo, ciscoBtnMat);
    ciscoBtn.position.set(offsetX + frameW/2 + spacing/2, baseY + buttonYOffset,
baseZ);
    ciscoBtn.rotation.y = Math.PI;
    ciscoBtn.cursor = 'pointer';
    scene.add(ciscoBtn);
    // Взаємодія: клік по кнопках відкриває відповідні сайти
    const raycaster = new THREE.Raycaster();
    const mouse = new THREE.Vector2();
    function onClick(event) {
      mouse.x = (event.clientX / window.innerWidth) * 2 - 1;
      mouse.y = -(event.clientY / window.innerHeight) * 2 + 1;
      raycaster.setFromCamera(mouse, camera);
      const intersects = raycaster.intersectObjects([oracleBtn, ciscoBtn]);
      if (intersects.length > 0) {
        if (intersects[0].object === oracleBtn) {
          window.open('https://academy.oracle.com/', '_blank');
        } else if (intersects[0].object === ciscoBtn) {
          window.open('https://www.netacad.com/', '_blank');
        }
      }
    }
    window.addEventListener('click', onClick);
  })();
```

```
  // --- Чат через WebSocket (працюючий, з іменем) ---
  let ws;
  let playerName = localStorage.getItem('playerName') || ('Гість_' +
Math.floor(Math.random()*1000));

  // --- Прибрано блок чату та весь код WebSocket-чату ---

  // --- МУЛЬТИПЛЕЄР: синхронізація гравців через WebSocket ---
  let playerId = Math.random().toString(36).substr(2, 9); // простий унікальний id
  let players = {}; // {id: {x, y, z}}
  let playerMeshMap = {}; // {id: mesh}

  function sendPlayerPosition() {
    if (ws && ws.readyState === 1) {
      ws.send(JSON.stringify({
        type: 'player',
        id: playerId,
        pos: { x: camera.position.x, y: camera.position.y, z: camera.position.z }
      }));
    }
  }

  function updateOtherPlayers() {
    for (const id in players) {
      let mesh = playerMeshMap[id];
      if (!mesh) {
        // Додаємо просту модельку (кулю) для кожного гравця
        const geometry = new THREE.SphereGeometry(2, 16, 16);
        // Свій гравець — зелений, інші — сині
```

```
        const color = (id === playerId) ? 0x43a047 : 0x1976d2;
        const material = new THREE.MeshStandardMaterial({ color });
        mesh = new THREE.Mesh(geometry, material);
        scene.add(mesh);
        playerMeshMap[id] = mesh;
       }
       mesh.position.set(players[id].x, players[id].y, players[id].z);
     }
     // Видалити модельки, яких вже немає
     for (const id in playerMeshMap) {
       if (!players[id]) {
        scene.remove(playerMeshMap[id]);
        delete playerMeshMap[id];
       }
     }
   }

   // Додаємо відправку позиції раз на 100мс
   setInterval(sendPlayerPosition, 100);



  </script>
</body>
</html>
```

Додаток Г

server.js

```
const http = require('http');
```

```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');

// --- Додаємо WebSocket для чату ---
const WebSocket = require('ws');
let wsServer;

function setupWebSocket(server) {
  wsServer = new WebSocket.Server({ server });
  wsServer.on('connection', function connection(ws) {
    ws.on('message', function incoming(message) {
      wsServer.clients.forEach(function each(client) {
        if (client.readyState === WebSocket.OPEN) {
          client.send(message.toString());
        }
      });
    });
  });
}

// --- Додаємо роздачу frontend ---
app.use(express.static(__dirname + '/../frontend'));

// --- Запуск сервера ---
const PORT = process.env.PORT || 3000;
const httpServer = require('http').createServer(app);
setupWebSocket(httpServer);
```

```
httpServer.listen(PORT, () => {
  console.log('HTTP/WebSocket сервер запущено на http://localhost:' + PORT);
});

// Ініціалізація SQLite
const sqlite3 = require('sqlite3').verbose();
const cors = require('cors');
const jwt = require('jsonwebtoken');
const bcrypt = require('bcrypt');

const port = 3000;
const JWT_SECRET = 'your_jwt_secret_key';

app.use(cors());
app.use(bodyParser.json());

const db = new sqlite3.Database('./database.sqlite', (err) => {
  if (err) {
    console.error('Error opening database:', err.message);
  } else {
    console.log('Connected to SQLite database');
    db.run(`
      CREATE TABLE IF NOT EXISTS USERS (
        ID INTEGER PRIMARY KEY AUTOINCREMENT,
        USERNAME TEXT UNIQUE NOT NULL,
        PASSWORD_HASH TEXT NOT NULL
      )
    `);
```

```
    db.run(`
      CREATE TABLE IF NOT EXISTS OBJECTS (
        ID INTEGER PRIMARY KEY AUTOINCREMENT,
        NAME TEXT NOT NULL,
        DESCRIPTION TEXT,
        URL TEXT NOT NULL
      )
    `);
  }
});

function authenticateToken(req, res, next) {
  const authHeader = req.headers['authorization'];
  const token = authHeader && authHeader.split(' ')[1];

  if (!token) return res.sendStatus(401);

  jwt.verify(token, JWT_SECRET, (err, user) => {
    if (err) return res.sendStatus(403);
    req.user = user;
    next();
  });
}

app.post('/register', async (req, res) => {
  const { username, password } = req.body;
  const hashedPassword = await bcrypt.hash(password, 10);
```

```
  db.run(
    `INSERT INTO USERS (USERNAME, PASSWORD_HASH) VALUES (?, ?)`,
    [username, hashedPassword],
    function (err) {
      if (err) {
        console.error(err.message);
        res.status(500).send('Registration failed');
      } else {
        res.status(201).send('User registered');
      }
    }
  );
});

app.post('/login', async (req, res) => {
  const { username, password } = req.body;

  db.get(
    `SELECT * FROM USERS WHERE USERNAME = ?`,
    [username],
    async (err, user) => {
      if (err) {
        console.error(err.message);
        res.status(500).send('Login failed');
      } else if (user && (await bcrypt.compare(password, user.PASSWORD_HASH)))
{
        const token = jwt.sign({ username: user.USERNAME }, JWT_SECRET, {
expiresIn: '2h' });
        res.json({ token });
```

```
      } else {
        res.status(401).send('Invalid credentials');
      }
    }
  );
});

app.post('/objects', authenticateToken, (req, res) => {
  const { name, description, url } = req.body;

  db.run(
    `INSERT INTO OBJECTS (NAME, DESCRIPTION, URL) VALUES (?, ?, ?)`,
    [name, description, url],
    function (err) {
      if (err) {
        console.error(err.message);
        res.status(500).send('Failed to save object');
      } else {
        res.status(201).send('Object metadata saved');
      }
    }
  );
});

app.get('/objects', (req, res) => {
  db.all(`SELECT * FROM OBJECTS`, [], (err, rows) => {
    if (err) {
      console.error(err.message);
```

```
      res.status(500).send('Failed to fetch objects');
    } else {
      res.json(rows);
    }
  });
});

// --- МУЛЬТИПЛЕЄР: синхронізація гравців через WebSocket ---
const players = {};

wsServer.on('connection', function connection(ws) {
  let thisId = null;
  ws.on('message', function incoming(message) {
    let data;
    try { data = JSON.parse(message); } catch { data = null; }
    if (data && data.type === 'player') {
      thisId = data.id;
      players[thisId] = data.pos;
      // Розсилаємо всім актуальні позиції
      const payload = JSON.stringify({ type: 'players', players });
      wsServer.clients.forEach(function each(client) {
        if (client.readyState === WebSocket.OPEN) {
          client.send(payload);
        }
      });
    } else if (data && data.type === 'chat') {
      // Новий чат: розсилаємо з іменем
      const payload = JSON.stringify({ type: 'chat', name: data.name, text: data.text });
```

```
      wsServer.clients.forEach(function each(client) {
        if (client.readyState === WebSocket.OPEN) {
          client.send(payload);
        }
      });
    } else if (typeof message === 'string') {
      // старий чат (fallback)
      wsServer.clients.forEach(function each(client) {
        if (client.readyState === WebSocket.OPEN) {
          client.send(message.toString());
        }
      });
    }
  });
  ws.on('close', function() {
    if (thisId && players[thisId]) {
      delete players[thisId];
      // Оновити всім список гравців
      const payload = JSON.stringify({ type: 'players', players });
      wsServer.clients.forEach(function each(client) {
        if (client.readyState === WebSocket.OPEN) {
          client.send(payload);
        }
      });
    }
  });
});
```